



Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**
Institut für Datenbanken
und Informationssysteme

Adaptierung und Neugestaltung einer Android Anwendung unter Berücksichti- gung der Material Design Richtlinien

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Florian Hofherr

florian.hofherr@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Johannes Schobel

2015

Fassung 7. März 2016

© 2015 Florian Hofherr

This work is licensed under the Creative Commons. Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF- \LaTeX 2_ε

Abstract

Es gibt viele Anwendungen, die technisch noch den Ansprüchen der Schöpfer genügen, deren Oberfläche jedoch nicht mehr zeitgemäß ist. Das alte Design folgte bereits einigen Richtlinien. Dabei galt es, einen Kompromiss aus den eigenen Anforderungen an die Anwendung und denen der Design-Richtlinien zu finden. Dieses Grundprinzip ändert sich nicht, aber der technische Stand, auf den es angewendet wird. So heißt es in der Physik „*Actio est reactio*.“, auf eine Aktion folgt immer eine Reaktion. Genau diesen Grundsatz vertreten auch die Material Design Guidelines. Mit Android Lollipop Version 5.0 kam dieses neue Design auf. Und als Entwickler für Android-Applikationen gilt es, dieses neue Design umzusetzen. Ob und wie praktikabel die daraus resultierenden Richtlinien, festgehalten in den Material Design Guidelines, tatsächlich sind, soll in dieser Arbeit überprüft werden. Zu diesem Zweck wird eine bereits bestehende Applikation zur mobilen Datenerhebung adaptiert und neugestaltet. Dabei wird unter Beachtung der speziellen Anforderungen dieser Applikation das neue Design entwickelt und mithilfe der Material Design Guidelines – respektive Meinungen Dritter – unter Android implementiert.

Danksagung

Ich danke allen innerhalb und außerhalb der Universität, allen Beteiligten und Betroffenen, für Ihre Geduld. Außerdem Herrn Professor Manfred Reichert, da er mich zügig an meinen Betreuer, Johannes Schobel, vermittelt hat. Meinem Betreuer danke ich insbesondere für die ambitionierte und kontinuierliche Rücksprache. Und letzten Endes bin ich auch dankbar für meine eigene Geduld.

Inhaltsverzeichnis

1	Einleitung	1
2	Material Design	5
2.1	Leitgedanken des Material Designs	6
2.2	Realisierung des Material Designs	7
3	Anforderungen	11
3.1	Nicht-funktionale Anforderungen	12
3.2	Funktionale Anforderungen	14
4	Konzeption und Planung	19
4.1	Splashscreen	20
4.2	Startseite	21
4.3	Fragebogenelemente	25
5	Umsetzung und Implementierung	31
5.1	Ressourcenmanagment	32
5.2	Startseite	36
5.3	Fragebogenelemente	40
6	Anforderungsabgleich und Diskussion	51
7	Zusammenfassung und Ausblick	57
7.1	Zusammenfassung	57
7.2	Ausblick	59

1

Einleitung

Diese Arbeit beschäftigt sich damit, ob die Adaptierung und Neugestaltung einer Anwendung mithilfe eines Design Katalogs sinnvoll ist. Insbesondere geschieht dies hier im Rahmen des offiziellen Material Design Katalogs bzw. der darin enthaltenen Guidelines. Das Problem dabei ist, dass die Einhaltung der sich stets verändernden und entsprechend schnell wieder veralteten Standards einer Plattform viele Änderungen mit sich bringen.

Nutzern ist bereits das ständige Aktualisieren von Anwendungen ein Dorn im Auge, geschweige denn von den sich teilweise stark verändernden Oberflächen. Natürlich möchten die Ersteller solcher offiziellen Guidelines das Erlebnis des Nutzers auf ihrer Plattform optimieren und vereinheitlichen. Allerdings stellt sich die Frage, wie konsequent diese Guidelines eingehalten werden sollten und wie aussagekräftig diese sind.

1 Einleitung

Des Weiteren ergibt sich die Problemstellung, dass das, was in den Guidelines gefordert wird, mit den gegebenen Mitteln nicht sauber zu implementieren ist. Sei es durch Fehler oder weil diese Mittel erst nachgereicht werden. Darum muss man sich auch Gedanken machen, welchen Kompromiss man eingehen möchte zwischen einer schnellen Anpassung an die aktuellen Guidelines und der Prämisse, dass damit Fehler einhergehen. Oder aber, dass man durch die geringere Unterstützung in der Anfangsphase einer Neuerung einen Mehraufwand hat, der zu einem späteren Zeitpunkt, zu dem einige Fehler bereits behoben wurden, nicht auftreten würde.

Im Hinblick auf das zuletzt genannte Problem wäre ein Lösungsansatz, für sogenannte *Minor Releases* zunächst zu entscheiden, wie man vorgehen könnte. Das Ergebnis hält man fest und stellt einen Plan auf, den man mit jedem weiteren Minor Release aktualisiert. Allgemein hätte man dann bei einem bevorstehenden *Major Release* einen Plan zur Neugestaltung vorliegen, den man dann umsetzen könnte. Der Vorteil daran wäre auch, dass Fehler und Schwierigkeiten der Minor Releases bereits teilweise behoben wurden, es schon ein paar Implementierungen gibt und man im Hinblick auf die gesamte Struktur der Applikation fundierte Entscheidungen treffen kann. Dadurch würden letzten Endes sowohl der Mehraufwand als auch die Fehlerwahrscheinlichkeit reduziert, und das Benutzererlebnis sowie die Wartbarkeit würden dadurch weniger darunter leiden.

Nicht zuletzt kann es auch denkbar, dass Änderungen sich nicht durchsetzen, widerrufen, oder auch nicht weiter unterstützt werden. Und zuletzt besteht die Möglichkeit im Einzelnen zu entscheiden, ob man eine mögliche Library findet und verwendet, oder sich selbst den Implementierungsaufwand auflädt, weil es nötig ist, um die eigenen Anforderungen ausreichend zu erfüllen. Daher betrachten wir im folgenden Kapitel zunächst, worum es beim Material Design geht, was die Intention der Schöpfer war und worauf diese Beschlüsse beruhen. Danach und im Weiteren dieser Arbeit wird eine Applikation für die Darstellung von Fragebögen umgestaltet und anhand dessen eine Entscheidung getroffen, wo, ob und wieso der Design Guide geeignet oder ungeeignet ist.

Der Aufbau der Arbeit gliedert sich wie folgt: Als erstes kommt eine kurze Einführung in das Material Design im nachfolgenden Kapitel 2. Darauf folgt mit Kapitel 3 die Aufnahme der Anforderungen und danach in Kapitel 4 die Konzeption und Planung der Umgestal-

tung. Dabei gehen wir auf die Ansprüche des Design Guides ein. Dieser rote Faden zieht sich durch die gesamte Arbeit und mündet schließlich in einer Bewertung des Design Guides und der fundierten Erläuterung von Abweichungen von diesem. Insbesondere Kapitel 4 und Kapitel 5 stehen hier miteinander in Kontrast, welcher zusammen mit den aufgestellten Anforderungen zu den Ergebnissen in Kapitel 6 wesentlich beitragen wird. Kapitel 7 schließt die Arbeit mit einer Zusammenfassung und Ausblick auf weitere Arbeiten ab.

2

Material Design

„We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design. This spec is a living document that will be updated as we continue to develop the tenets and specifics of material design.“ [1, Material properties]

Dieses Zitat bedeutet so viel, wie, dass Material Design gute Designprinzipien mit der Innovation und Möglichkeit von Technik und Wissenschaft vereinen soll. Aber auch, dass diese Spezifikation, in der das Material Design definiert wird, sich stetig verändern wird. Angekündigt Mitte 2014 und eingeführt 2015, werden nach und nach zumindest die Google-eigenen Applikationen darauf umgestellt. Es soll ein Fundament entwickelt werden, welches allgemeine Funktionen anbietet. Dazu zählen Aspekte des Oberflächendesigns, der Interaktion mit dieser und der Benutzererfahrung mit beidem. Darunter

2 Material Design

verbergen sich unterschiedliche Kategorien, wie Dialoge, Listen oder Navigationselemente. Und damit wird eine geräte- und plattformübergreifende Vereinheitlichung der Benutzererfahrung angestrebt. D.h. das Material Design richtet sich somit nicht nur an Android, sondern alle Plattformen. Allerdings betrachten wir im Kontext dieser Arbeit nur Aspekte der Guidelines, die sich auf Android beziehen. Und insbesondere Aspekte, die sich auf Tablets beziehen. Außerdem werden nur Grundgedanken und relevante Aspekte betrachtet, die dazu nötig sind, diese Arbeit besser einordnen und nachvollziehen zu können.

2.1 Leitgedanken des Material Designs

Ein Grundgedanke des Material Designs ist, metaphorisch die Realität abzubilden. Es orientiert sich stark an Tinte und Papier und hat den Anspruch, möglichst realistische Effekte zu erzeugen. Sowohl statische Optik als auch Bewegung werden abgebildet. Dazu werden Effekte wie Licht herangezogen. Insgesamt soll dieses Konzept somit zu einer intuitiven und angenehmen Wahrnehmung beitragen. Ein weiterer Hintergrund des Material Designs ist, dass durch den angestrebten Realismus Betonung auf bestimmte Aktionen gelegt werden kann. So soll der Benutzer eine fließende Umgebung vor sich haben und subtil von dem Design geführt werden. Gleichzeitig soll durch realistisch anmutende Bewegungen als Reaktion auf eine Aktion der Fokus des Benutzers neu gesetzt werden. Dadurch wird der Blick des Benutzer auf wichtige Ereignisse gelenkt und es kann eine Hierarchie abgebildet werden. Bewegung soll dabei immer eine Bedeutung beigemessen werden und Feedback auf Aktionen soll zurückhaltend und zugleich deutlich gegeben werden. Außerdem sollen Übergänge schlüssig und effizient sein, also keine überflüssigen Ausschmückungen beinhalten und stetig ablaufen. Unter stetig kann man sich vorstellen, dass runde, harmonische Bewegungen ausgeführt werden. In der physikalischen Welt sind abrupte positive wie auch negative Beschleunigungen nicht anzutreffen. Das würde einem Auto gleichen, das aus voller Fahrt sofort regungslos stehen bliebe.

Im nächsten Kapitel werden konkrete Realisierungen genannter Gedanken betrachtet.

2.2 Realisierung des Material Designs

Ein erster Aspekt der Realisierung der Leitgedanken aus Kapitel 2.1 ist, dass eine neue Komponente für das Oberflächendesign eingeführt wird. Typischerweise hat reales Material drei Dimensionen, also eine räumliche Darstellung. Um das zu erreichen, wird zum 2D-Koordinatensystem des Bildschirms noch eine weitere Komponente, eine z-Achse, hinzugefügt (siehe Abbildung 2.1). Im Weiteren bezeichnet der Begriff „Material“ den bildhaften Stoff, aus dem das Material Design aufgebaut ist.



Abbildung 2.1: 3D-Raum des Material Designs [1, Introduction]

Darauf baut konsequent das Spiel aus Licht und Schatten auf. Auf Android Geräten werden Schatten mithilfe virtueller Lichtquellen erzeugt. Dabei gibt es Schlüssel-Lichter (*en. key lights*), welche gerichtete und etwas schärfere Schatten werfen, und Umgebungslicht (*en. ambient light*), das allgemeineren, ungerichteten Schatten emuliert. Schatten, der vom Umgebungslicht erzeugt wird, ist eher marginal und weich. Dieser fällt außerdem überall nieder und nicht nur an spezifischen Kanten. Im Einzelnen lässt sich diese Vorgehensweise in Abbildung 2.3 betrachten. Man orientiert sich dabei an der z-Koordinate, die direkten Einfluss auf den Schattenwurf hat. So wird z.B. auch kein Schatten geworfen, wenn keine Differenz zweier Elemente in der z-Koordinate vorliegt, da diese sozusagen auf derselben Ebene liegen und eine glatte Fläche bilden. Ist der z-Wert höher, so wird vom *höheren* Element ein Schatten auf das *darunter* liegende Element

2 Material Design

geworfen. Daraus lässt sich bereits erkennen, dass die Erhebungen der Elemente relativ zueinander erfolgen.

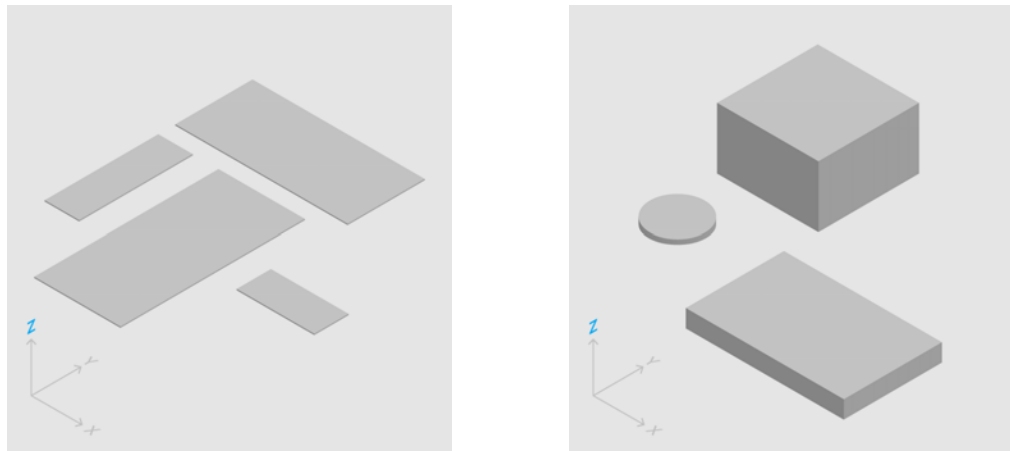


Abbildung 2.2: Links korrekt mit 1dp Dicke und rechts falsch mit mehr als 1dp Dicke [1, Environment]

Allerdings soll Material immer eine Dicke von 1dp (density-independent pixel – dichteunabhängiger Pixel) erhalten. Wohingegen natürlich die x- und y-Komponenten variieren dürfen, wie in Abbildung 2.2 zu sehen ist [1, Elevation and Shadows]. In dieser Maßeinheit wird in der Regel alles gemessen, da sie für Geräte mit unterschiedlicher Pixeldichte funktionieren und so sowohl die konsistente Darstellung als auch die Erstellung von Oberflächen für verschiedene Geräte erleichtern. Die Pixeldichte ist nicht zu verwechseln mit der Auflösung eines Bildschirms. Auflösung und Pixeldichte sind zwei voneinander verschiedene Messgrößen. Wobei die Pixeldichte die ppi-Zahl misst, also wie viele **pixel per inch** sich auf einem Bildschirm befinden. Die Auflösung hingegen beschreibt die insgesamt Anzahl von Pixel, die ein Bildschirm besitzt, und gegebenenfalls auch das Seitenverhältnis desselben. Inhalt, der auf solchen Flächen dargestellt wird, leistet übrigens keinen Beitrag zur Dicke der Fläche, und ist durch diese Fläche begrenzt – ganz so als würde man auf ein Blatt Papier zeichnen [1, Principles].

Gerne wird der Schatteneffekt zum metaphorischen Anheben eines Elements genutzt. Ein Beispiel dafür stellen Buttons dar, die angeklickt oder fokussiert werden. Aber auch Elemente, die *über* andere hinweg bewegt werden. Beispiel hierfür wäre ein Drag-and-Drop-Szenario, in dem man eine Liste von Elementen manuell umsortieren kann. Dabei

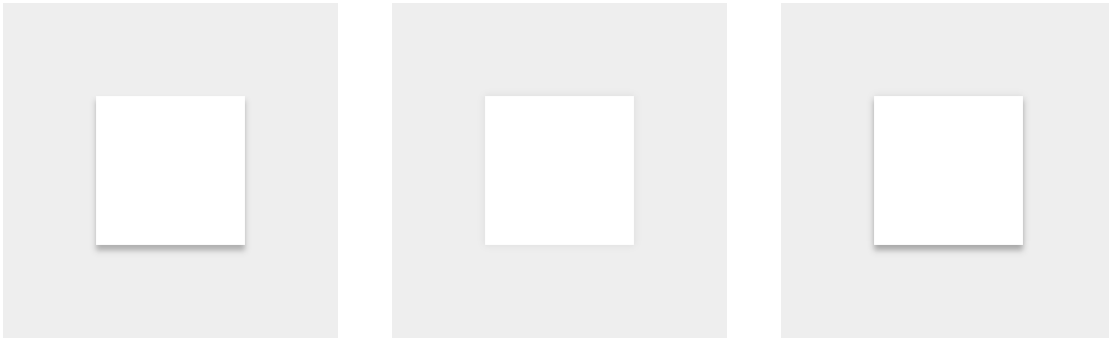


Abbildung 2.3: Schattenwurf, von links nach rechts: mit Schlüssel-Licht, mit Umgebungslicht, kombiniert [1, Elevation and Shadows]

gilt, dass Material solide ist und sich somit Material nicht überschneidet oder durchlässig für anderes Material ist. Trotzdem kann Material seine Form verändern. Dazu gehören z.B. Größe oder Geometrie des Materials [1, Environment]. In Kombination kann man somit schön und verständlich Elemente einbauen. Unterschiedliche Erhebungen in der z-Richtung je nach Lage eines Elements innerhalb eines sich verändernden Layouts auf dem Bildschirm schafft so mehr Ordnung und hilft, Elementen eine bestimmte Rolle zuzuordnen.

Ein anderer Aspekt ist, dass das Erstellen von neuem Material, also neuen Elementen, in der Regel einen Ausgangspunkt hat, von dem es sichtbar ausgehend ausbreitet, statt einfach plötzlich auf dem Bildschirm zu erscheinen. Hier kommen die bereits genannten Übergänge ins Spiel. Bei diesen sollen klare Linien eingehalten werden und der Aufbau eines Layouts einem Trend folgen. So soll beispielsweise eine Kachelfläche oder eine Liste von Elementen in einer bestimmten Reihenfolge erscheinen, anstatt auf einmal angezeigt zu werden. Der Benutzer kann somit visuell geführt werden. Auch minimale Bewegungen, z.B. in Icons, die sich im Ruhezustand oder auch bei Aktivität optisch verändern, sind denkbar.

Ein weiteres beliebtes Designelement stellt die Reaktion von Oberflächen auf Benutzerinteraktionen dar. Das System stellt dann sofortiges visuelles Feedback analog zum eingetroffenen Ereignis bereit. So z.B. einen sogenannten Ripple-Effekt, wenn man einen Button klickt oder fokussiert. Dabei wird ausgehend vom Kontaktpunkt die Fläche kurzzeitig, bzw. für die Dauer des Kontaktes, anders gefärbt. Üblich ist dabei die radiale

2 Material Design

Ausbreitung innerhalb der betroffenen Fläche, analog einem Stein, den man in einen ruhigen See wirft.

Auf Details wie Farben, Layouting und dergleichen wird an passender Stelle im Verlauf der Arbeit eingegangen. Auch Farben richten sich nach realen Vorbildern. Als Inspiration werden unter anderem Sportplätze angegeben, gemeint sind vermutlich die bunten Laufbahnen („*athletic courts*“ [1, Color]).

3

Anforderungen

In diesem Kapitel werden die nicht-funktionalen und funktionalen Anforderungen beschrieben und erläutert, auf die bei dieser Arbeit in Hinblick auf die Entwicklung der zugrunde liegenden Applikation geachtet wurden.

Da das Hauptaugenmerk lag darauf, die bereits bestehende Applikation unter Verwendung des Google Material Design Katalogs umzugestalten, können und müssen einige grundlegende Anforderungen aus der vorherigen Arbeit [2] übernommen werden. Allerdings beschränkt sich die hier vorliegende Arbeit auf die Neugestaltung der allgemeinen Benutzerseite und lässt den Administrationsbereich größtenteils außen vor.

Außerdem fallen die Anforderungen insgesamt eher generisch aus, was sich prinzipiell auch auf die praktische Umsetzung auswirkt. Als Zielgeräte der Anwendung sollen vorrangig Tablets statt Smartphones betrachtet werden. Das liegt unter anderem an den

hohen Aufwänden zur Gewährleistung der konsistenten Darstellung auf den verschiedenen Hardwaresystemen – insbesondere bezüglich Bildschirmgrößen.

3.1 Nicht-funktionale Anforderungen

Bei den nicht-funktionalen Anforderungen handelt es sich um Ansprüche bezüglich der Qualität und allgemeine Funktionalitäten, anstatt um konkrete Funktionen und Elemente.

N1 Aussehen und Handhabung: Das Aussehen und die Handhabung sollen sich am Material Design orientieren. Das heißt im Klartext, dass sich das neue Design der Applikation *anfühlen* soll, wie die verwendete Plattform – in diesem Fall Android Lollipop. Dazu gehören neben der Optik auch die Funktionen, also Dinge wie Interaktions- und Reaktionsmuster. Was wiederum bedeutet, dass die verwendeten Elemente möglichst solche sein sollen, die Android vorsieht. Außerdem sollen sich verwendete Elemente in einem Kontext wiederfinden, wie ihn Android definiert und er im zugehörigen Material Design Katalog beschrieben wird.

N2 Benutzbarkeit: N1 steht in direkter Korrelation mit der Benutzbarkeit. Die Benutzer sollen sich intuitiv zurecht finden. Die Bedienbarkeit soll dem Benutzer insofern entgegenkommen, als dass er sich in der Ausführung einer Aufgabe oder dem Verfolgen eines Ziels unterstützt fühlt. Idealerweise soll die Bedienoberfläche eine flüssige Interaktion ermöglichen, bei der alle Oberflächenelemente des Fragebogens so reagieren, wie der Benutzer es erwarten würde und benötigt. Dieses Verhalten ist natürlich nur näherungsweise erreichbar, da jeder Benutzer individuelle Vorstellungen haben kann, die nicht zwingend deckungsgleich sein müssen. Es gilt also eine gute Schnittmenge zu finden. Zudem müssen einige individuelle Aspekte betrachtet. Unter anderem Bedürfnisse älterer Menschen [3].

N3 Gestaltung: Zwei weitere wichtige Aspekte sind sowohl die Farbgestaltung, als auch das generelle Design der Applikation. Insbesondere bei Farben muss auf Einschränkungen (z.B. Sehschwächen [4, 5]) mancher Benutzer geachtet werden, und in Bezug auf den Fragebogen an sich auch auf den psychologischen Einfluss in der Gestaltung von Elementen. So sollten keine Rot-Grün-Farbkombinationen

verwendet werden oder auch schwache Kontraste, durch die Fragestellungen oder Antwortmöglichkeiten nicht gut lesbar erscheinen. Elemente zur Auswahl von Antworten sollten durch die voreingestellte Auswahl keinen Einfluss auf die Entscheidung des Benutzers haben. D.h. der Fragebogen muss eine möglichst neutrale Wirkung auf den Benutzer erzielen, damit das Ergebnis nicht verfälscht wird.

N4 Zuverlässigkeit: Da es sich bei Fragebögen um eine einmalige Datenerhebung handelt, müssen die Daten nach der Eingabe stets zuverlässig und persistent gespeichert werden.

N5 Performance: Wie bereits in N2, ist eine flüssige Interaktion gewünscht. Das bedeutet für die Leistung und Effizienz, dass die Oberfläche auf Benutzerinteraktionen möglichst schnell reagiert. Beispielsweise muss auf einen Klick auf ein Button-Element unmittelbar eine für den Akteur sichtbare Reaktion folgen. In diesem Zusammenhang kann das als Touchfeedback bezeichnet werden. Dadurch soll auch die Interpretation als Fehlverhalten vermieden werden, was ansonsten der Benutzbarkeit schadet.

N6 Betrieb und Umgebungsbedingungen: Das Zielsystem ist natürlich auch der Ursprung des Design Katalogs – nämlich Android Lollipop. Außerdem ist, auch wenn eine Beschränkung auf Tablets vorliegt, wünschenswert, dass auf vielen verschiedenen Geräten eine korrekte Darstellungen der generierten Fragebögen ermöglicht wird.

N7 Abwärtskompatibilität: Des Weiteren ist auch eine Abwärtskompatibilität insofern erwünscht, dass die Applikation auf Android KitKat (Version 4.4) läuft und verwendet werden kann. Das ist jedoch nur eine sekundäre Zielsetzung und der Designanspruch dafür geringer. Das Hauptaugenmerk bleibt auf Android Lollipop.

N8 Wartbarkeit und Änderbarkeit: Die Wartbarkeit und Änderbarkeit bezieht sich darauf, dass die Applikation möglichst modular aufgebaut ist. Die entsprechenden Oberflächenelemente sollen flexibel verwendet, leicht um weitere Elemente ergänzt oder verändert werden können. Das ergibt vor allem auch in Hinsicht auf die Verwendung Sinn, da man generische Fragebögen erzeugen möchte.

3 Anforderungen

Fest vorprogrammierte Fragebögen benötigen bei jeder Änderung eine manuelle Anpassung.

N9 Portierbarkeit und Übertragbarkeit: Eine Portierbarkeit im eigentlichen Sinne ist zwar nicht explizit geplant, jedoch soll der Teil der Oberflächenerstellung für den Fragebogen später in einen *EBP (Executable Business Process)* integriert und dort verwendet werden.

N10 Korrektheit: Selbstredend, aber der Vollständigkeit halber hier auch aufgeführt, sollen Eingaben der Benutzer korrekt verarbeitet und angezeigt werden.

3.2 Funktionale Anforderungen

In diesem Abschnitt geht es um funktionale Anforderungen, in welchen konkrete Funktionen und Elemente des Systems beschrieben werden. Dabei sollen natürlich Funktionen aus der zugrunde liegenden Arbeit [2] möglichst erhalten bleiben.

F1 Mehrsprachigkeit der Bedienoberfläche: Die Mehrsprachigkeit bezieht sich auf zwei verschiedene Aspekte (F1 und F2). Einerseits ist damit die Sprache der Applikation selbst gemeint, also beispielsweise die Menüführung, Buttons und dergleichen. Diese soll verändert werden können und leicht und eindeutig zugänglich gemacht werden, da diese Funktion aller Wahrscheinlichkeit nach oft verwendet wird. Davon ist überwiegend der Teil der Benutzer betroffen, der den Fragebogen anbietet und mittels der Applikation bereitstellen, bzw. auswerten möchte.

F2 Mehrsprachigkeit der Fragebogenoberfläche: Andererseits ist damit aber auch die Sprache der Fragebögen selbst gemeint, die unabhängig von der Bedienoberfläche regulierbar sein soll. Anzumerken ist, dass zum Fragebogen Bedienelemente gehören, wie normale Kontrollelemente oder Editoren. Allerdings werden diese im Rahmen des Fragebogens betrachtet, wohingegen sich die Bedienoberfläche rein auf die vorangehenden Kontrollelemente bezieht. Diese getrennte Betrachtungsweise hat seinen Sinn darin, dass der andere Teil der Benutzer – also Teilnehmer, die mittels der Applikation befragt werden sollen – nicht zwangsläufig

dieselbe Sprache beherrscht, oder die jeweiligen Parteien eine andere Sprache bevorzugen.

F3 Dynamische Generierung: Die dynamische Generierung eines Fragebogens innerhalb der Applikation soll auf Grundlage einer Fragebogenvorlage möglich sein. Dazu wird in einer anderen Anwendung aus den Elementen, die die Applikation realisiert, beliebig ein Fragebogen erstellt. Dieser soll dann auf das Gerät übertragen und dort mit den gelieferten Informationen erzeugt werden.

F4 Fragebogenelemente: Grundsätzlich bleiben die wesentlichen funktionalen Anforderungen an die Benutzeroberfläche des Fragebogens bestehen, wie sie bereits in der zugrunde liegenden Arbeit [2] definiert wurden. Im Detail wird in den folgenden Kapiteln auf die einzelnen Elemente der Oberfläche eingegangen. Hinzu kommen neben der Umgestaltung für Android Lollipop auch neue Aspekte. Es soll beispielsweise für den Benutzer zu erkennen sein, ob ein Element (z.B. eine Frage im Fragebogen) obligatorisch oder optional ist.

F4.1 Überschrift und Textabschnitt: Die einfacheren Elemente sind die Überschrift, sowie der Text – mit oder ohne Paragraph. Ersteres Element soll, wie der Name schon vermuten lässt, lediglich eine Überschrift anzeigen. Das Text-Element hingegen sollte bisher einfach nur einen Textblock anzeigen mit der Option einen Paragraphen anzugeben oder nicht. Hinzu kam die neue Anforderung, dass der Text nun im Blocksatz formatiert dargestellt werden soll.

F4.2 Medien-Element: Neu hinzukommen soll ein Medien-Element, das ein Bild mit optionalem Untertitel darstellen soll.

F4.3 Eingabefelder und Editoren: Des Weiteren gibt es Eingabefelder (Text-, Nummern-, und Datumseingabe), die sich auch zusammenfassen lassen. Es soll der jeweilige Datentyp korrekt und passend angezeigt werden, bzw. zunächst ein passendes Eingabeelement – oder auch Editor (z.B. entsprechende Tastaturen oder ein Kalender) – dem Benutzer zur Verfügung gestellt wird. Die einzelnen Datentypen, die durch die Eingabefelder angezeigt werden, sollten selbsterklärend sein.

F4.4 Zweistellungs- und Dreistellungsschalter: Bereits etwas komplexer sind die Zweistellungs- und Dreistellungsschalter. Diese Elemente bieten zunächst die

3 Anforderungen

Möglichkeit über einen Slider zwischen zwei oder drei Antwortmöglichkeiten zu wählen. Im Wesentlichen geht es dabei um klassische Ja-Nein-Fragen, weshalb sich die Möglichkeiten auf *ja* und *nein* beschränken lassen. Per Voreinstellung soll die Auswahl bei diesen beiden Elementen neutral sein, um, wie schon in *N3* erläutert, keinen Einfluss auf die Entscheidung des Benutzers zu erwirken. Der Unterschied beider Elemente liegt dabei darin, dass beim Zweistellungsschalter die neutrale Position keine Antwortmöglichkeit ist, beim Dreistellungsschalter hingegen eine valide Option darstellt. Das bedeutet, dass beim Zweistellungsschalter die initiale neutrale Position nach einer Auswahl auch nicht mehr auswählbar sein soll.

F4.5 Mehrstellungsschalter: Will man mehrere Antwortmöglichkeiten zur Auswahl stellen, so soll ein Mehrstellungsschalter mit neutraler Position zur Verfügung gestellt werden.

F4.6 Bereichsauswahl: Außerdem soll es eine Bereichsauswahl bestehend aus einem Editor zur Eingabe und einem Anzeigeelement für die Eingabe geben. Damit soll der Benutzer aus einem gegebenen Bereich Teilbereiche auswählen können.

F4.7 Einfachauswahl: Die Einfachauswahl soll eine simple Auswahl einer Antwort aus einer gegebenen Anzahl von Antwortmöglichkeiten realisieren.

F4.8 Mehrfachauswahl: Eine Mehrfachauswahl ist eine Gruppe aus mehreren Zweistellungs- und Dreistellungsschaltern, bzw. noch einem Eingabefeld für Sonstiges, in das der Benutzer eine alternative Antwortmöglichkeit eintragen kann. Davon unterschieden wird ein *Frageblock*, bei dem beliebige Elemente in einen Block zusammengefasst werden können, die nicht voneinander abhängig sind. Das spielt für die spätere Validierung eine Rolle. Alternativ soll noch eine *einfache Mehrfachauswahl* zur Verfügung stehen, die eine simple Selektion mehrerer Auswahlmöglichkeiten realisiert, ohne dem einen gesonderten Wert zuzuweisen. Also das Verhalten, wie es z.B. eine Checkbox realisiert.

F4.9 Validierung: Zuletzt ist eine Validierung gefordert, ob alle obligatorischen Fragebogenelemente abgearbeitet wurden. Dadurch soll das Starten der nächsten

Fragebogenseite gesteuert werden, sodass der Start der Seite erst möglich ist, sobald alle zwingend notwendigen Antworten vorliegen.

F5 Menüführung: Die Menüführung bezieht sich auf alle Bedien- und Kontrollelemente der Applikation, die sich außerhalb der Fragebogenoberfläche befinden.

F5.1 Navigation: Für die Menüführung soll eine verbesserte Navigation implementiert werden, die Einstellungen und Navigationsmöglichkeiten aufnimmt und besser gliedert.

F5.2 Impressum: Das bisher vorhandene Impressum muss den anderen Änderungen angepasst werden, und soll zum Gesamteindruck der Applikation passen.

F5.3 Login: Mit der Administrationsseite an sich befasst sich diese Arbeit zwar nicht. Da sie aber Teil der zugrunde liegenden Arbeit ist und Bestandteil der Applikation, ist auf der allgemeinen Benutzerseite der Bedarf einer Anmelde-möglichkeit gegeben. Daher soll eine entsprechende Oberfläche realisiert werden, an die angeknüpft werden kann. Die Logik selbst ist nicht Teil dieser Arbeit.

F5.4 Fragebogenübersicht: Das bisherige Design der Fragebogenübersicht soll überarbeitet werden. Dies ist der neuen vereinfachten und überarbeiteten Navigation und der reduzierten Oberfläche geschuldet.

F5 Änderung der Bildschirmorientierung: Eine Änderung der Orientierung am Gerät soll unterstützt werden. Das bedeutet konkret, dass für die Applikation sowohl im *Portrait*- als auch im *Landscape-Mode* ein angemessenes Layout existiert.

Im nachfolgenden Kapitel wird zunächst die konzeptionelle Umsetzung der Anforderungen beschrieben. Dabei wird Bezug auf die einzelnen Anforderungen genommen und es werden einzelne, nennenswerte Elemente vorgestellt.

4

Konzeption und Planung

Dieses Kapitel befasst sich mit der Betrachtung der bereits bestehenden Applikation und der daraus resultierenden Konzeption und Planung zur Umgestaltung dieser in Bezug auf die neuen vorgegebenen Design Richtlinien. In Abbildung 4.1 sind die makroskopischen Segmente zu sehen und deren Übergangsmöglichkeiten ineinander. Zu den jeweiligen Segmenten werden die darin anzutreffenden Elemente genannt, gezeigt und erläutert – sowohl was den bisherigen Stand als auch den angestrebten Zustand betrifft. Dabei beschränken wir uns auf die grobe Zuordnung und Referenzierung der zugehörigen Einträge im Design Katalog, sowie der Erläuterung und Anwendung dieser Einträge. Außerdem werden nachfolgend im Wesentlichen nicht-triviale Elemente aufgeführt.

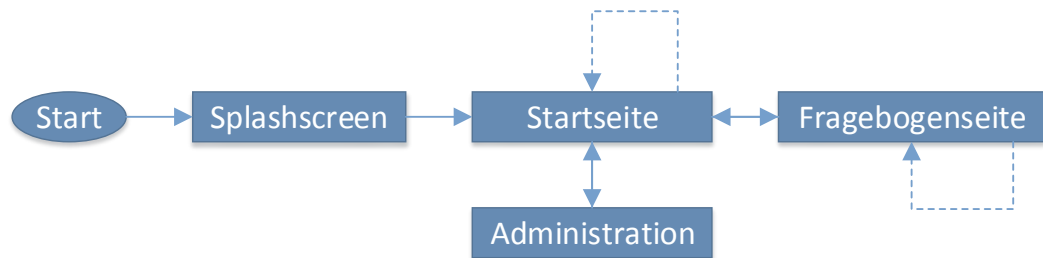


Abbildung 4.1: Vereinfachtes Ablaufdiagramm der mobilen Applikation

4.1 Splashscreen

Es gibt verschiedene Arten von Launchscreens die im Material Design Katalog [1, Launch screens] vorgeschlagen werden, genauer gesagt zwei. Launchscreens finden hauptsächlich beim Start der Applikation Anwendung. Der eine Typ wird als *Placeholder UI* bezeichnet, bei der statt eines leeren Bildschirms eine grobe, leere Vorschau der eigentlichen Oberfläche angezeigt wird, während im Hintergrund der Inhalt geladen wird. Die andere Möglichkeit ist der *Branded Launch*, bei dem, wie der Name schon sagt, das Augenmerk mehr auf der Marke liegt. Dazu wird z.B. ein Bild mit hohem Wiedererkennungswert verwendet. Allerdings finden auch Brandings und Logos Verwendung. Da der Splashscreen der bestehenden Applikation diese Angaben bereits erfüllt und sich farblich gut in das Konzept eingliedert, kann dieser unverändert beibehalten werden. Gegebenenfalls lässt sich dieser schnell abändern und anpassen. Außerdem macht ein solcher Launchscreen insofern Sinn, dass damit die gefühlte Wartezeit des Benutzers reduziert werden kann, die beim Verwenden eines leeren Bildschirms entstehen kann. Müssen vorab viele Daten abgerufen werden, bevor die Applikation vollständig geladen werden kann, ist es möglich mit dem Splashscreen die entstehende Ladezeit zu kaschieren.



Abbildung 4.2: Splashscreen im Ausgangszustand

4.2 Startseite

Die Startseite beinhaltet die Navigation und somit zählen die verschiedenen Seiten, die durch diese erreicht werden können, auch zum Kapitel 4.2.

Toolbar

Da laut Definition des Design Katalogs [1, App bar] eine Toolbar (oft auch als Appbar bezeichnet) zur Navigation, für Branding und für bestimmte Aktionen da ist, wird die alte Toolbar verworfen und eine neue erstellt. Dort wird das typische *Burger-Icon* ≡ auf der linken Seite zu sehen sein (siehe 4.4). Danach folgt der Seitentitel und rechts dann spezifische Aktionen innerhalb der App. Diese werden durch Icons, bzw. textuell im Dropdown-Menü, das geöffnet wird, sobald man das Menu Icon ganz rechts drückt, repräsentiert. Den Seitentitel als Titel in der Toolbar zu setzen wird auch vom Design Katalog vorgeschlagen. Die Sprachauswahl wird zudem als Dropdown realisiert und die Administration von der Toolbar entfernt und in die *Navigation* verlegt, da eher oft verwendete Optionen in die Toolbar gehören, um einen schnellen und gleichzeitig einfachen Zugriff zu ermöglichen. [1, Structure; Buttons; Menus; Toolbars]



Abbildung 4.3: Toolbar im Ausgangszustand



Abbildung 4.4: Toolbar-Konzept; Nav Icon oder auch Burger Icon (1) und Menu Icon (2)

Navigation

Im Ausgangszustand existierte lediglich eine Startseite, ohne konkrete Navigation. Bei einer großen Menge an Inhalt wird diese Konstellation allerdings zunehmend unübersichtlich, und weitere Einstellungen oder neue Inhalte sind schwer unterzubringen. Zudem nimmt das Impressum am Seitenanfang sehr viel Platz weg. Da außerdem eine Navigation alles verständlicher und übersichtlicher gestaltet, bietet es sich an, eine solche einzubauen. Zur Option stehen dabei Tabs oder ein sogenannter Navigation Drawer. Tabs sind jedoch weniger flexibel als ein Navigation Drawer.

Insbesondere in Hinsicht auf Erweiterbarkeit lohnt sich ein Navigation Drawer, da er mehr Platz für Elemente bietet und gegebenenfalls weitere Funktionalitäten implementiert werden können (z.B. ausklappbare Elemente und Überschriften für Subkategorien, die eine übersichtliche Strukturierung erlauben). Außerdem kann eine hierarchische Anordnung der Navigationsoptionen den Benutzer zusätzlich unterstützen, seinen Fokus auf wichtigere Punkte zu lenken. Laut Design Katalog eignet sich der Drawer bei Ap-

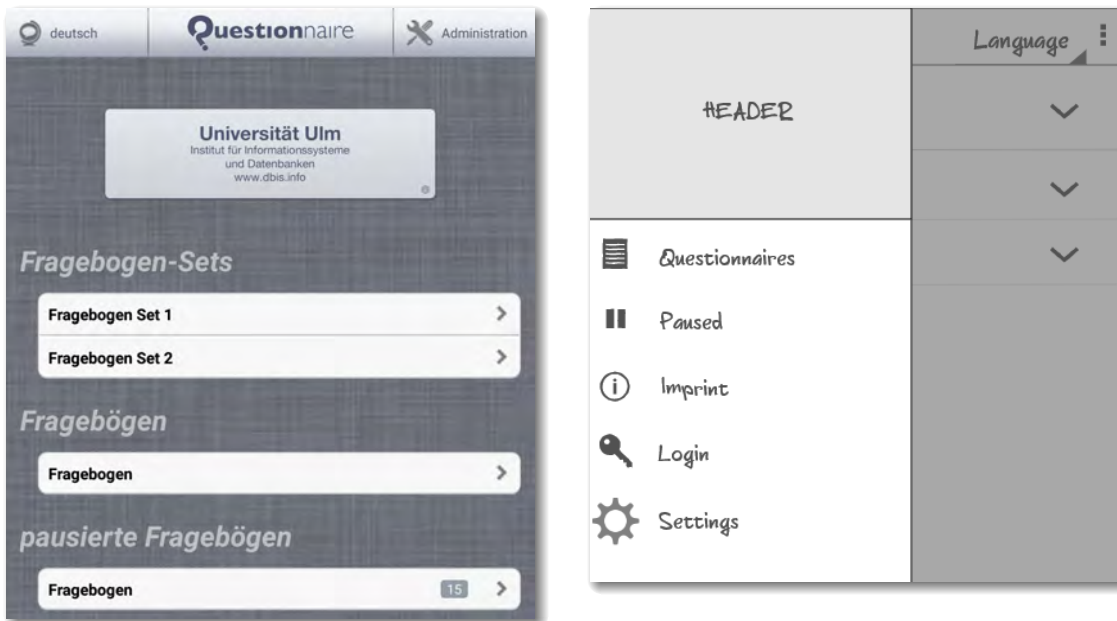


Abbildung 4.5: Startseite links im Ausgangszustand und rechts das neue Konzept

pplikationen mit vielen wichtigen Oberflächen, die nicht miteinander korrelieren, jedoch schnell zugreifbar sein sollen. Außerdem hilft er zur Reduzierung der Sichtbarkeit von weniger häufig genutzten Zielen und bei tieferen Navigationsstrukturen. Da möglicherweise noch weitere Navigationsoptionen hinzukommen können, ist es sinnvoll, mit einem solchen Navigation Drawer zu arbeiten. Das Design wird vom Design Katalog explizit vorgegeben. So stehen einige Maße und Elemente, wie z.B. der Seitenkopf und dessen Position und Maße, schon fest. Über die Navigationsoptionen im Navigation Drawer gelangt der Benutzer anschließend auf die jeweilige Seite. Dabei wird der Seitentitel aus Abbildung 4.4 neu gesetzt. [1, Navigation; Accessibility; Navigation drawer; Icons; Settings; Navigational transitions]

Fragebogenliste

Wie bereits in Kapitel 3.2 erwähnt, wird die Fragebogenliste reduziert und vereinfacht. Die Liste soll jedoch weiterhin aus ausklappbaren Listenelementen bestehen, wie es bereits in der ursprünglichen Implementierung der Fall war. Lediglich das Design wird geändert, damit es konform zum Design Katalog ist. Dazu gehört beispielsweise das

4 Konzeption und Planung



Abbildung 4.6: Fragebogenliste links im Ausgangszustand und rechts das neue Konzept

Icon, das auf der rechten Seite eines Listenelements der 4.6 zu sehen ist. Elemente innerhalb des ausgeklappten Listenelements werden entsprechend angepasst, können aber an sich bestehen bleiben.

Außerdem wird das Grunddesign der Liste abgeändert. Es wird keine Überschrift mehr geben – der Seitentitel aus Abbildung 4.4 erfüllt den Zweck bereits – und die Gruppierung durch die Container-Box mit abgerundeten Ecken wird aufgehoben. Damit wirkt das Design leichter und weniger überladen. [1, Lists; Lists: Controls]

Impressum

Das Impressum wird aus Platzgründen verschoben, um die Benutzeroberfläche übersichtlicher zu gestalten. Da es sich dabei lediglich um Textinhalte und Logos handelt, wird an dieser Stelle auf eine Visualisierung mittels Konzept verzichtet.

Login

Der Zugang zur Administration der Applikation findet in Form eines modalen Login-Dialogs statt. Da der Design Katalog dem nicht widerspricht, wird lediglich das Erscheinungsbild des Dialogs entsprechend angepasst. Dazu gehört z.B. das Abdunkeln des Hintergrundes, wenn dieser aufgerufen wird. [1, Dialogs; Text fields]



Abbildung 4.7: Login-Dialog links im Ausgangszustand und rechts das neue Konzept

4.3 Fragebogenelemente

Dieses Kapitel befasst sich mit den einzelnen Komponenten, aus denen ein Fragebogen aufgebaut werden soll.

Überschrift und Text

Bei der *Überschrift* handelt es sich um einen einfachen Text, der entsprechend gestaltet werden muss. Darunter fallen beispielsweise Schriftart, Textgröße und Textfarbe. Dabei bietet die Überschrift keinen optionalen Text an. Im Gegensatz dazu gibt es das *Textelement* mit obligatorischem Textfeld und optionaler Überschrift. Solche Elemente wie Text und Überschrift lassen sich trivial erstellen und anpassen. Lediglich die Darstellung des Textes in Blocksatz ist aufwändiger, da Android zwar rechts- und linksbündigen Text anbietet, jedoch keinen Blocksatz. Da im Material Design Katalog nichts zu finden ist, das für oder gegen eine Verwendung von Blocksatz spricht, kann dies allerdings anderweitig realisiert werden. [1, Typographie]

Eingabefelder

Eingabefelder für Texte, Zahlen und Datum, können alle durch normale Textfelder abgedeckt werden. Einem normalen Textfeld kann per Attribut ein Eingabetyp zugeordnet

4 Konzeption und Planung

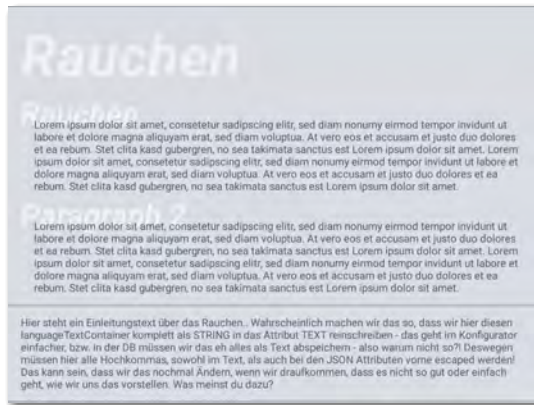


Abbildung 4.8: Überschrift und Text im Ausgangszustand

Abbildung 4.9: Eingabefelder im Ausgangszustand

werden. Der Button zum Löschen der Eingabe und die Titel einer Frage wiederum können übernommen werden, ebenso die Nummerierung der Fragen und die Anzeige, ob es sich um eine obligatorische Frage handelt. Lediglich das Design der einzelnen Elemente wie Textfelder und Button muss entsprechend angepasst werden. [1, Text fields]

Editoren

Mit Editoren sind die Eingabeelemente gemeint, die verwendet werden, um Eingaben in entsprechenden Eingabefeldern zu tätigen. Jedes der Eingabefelder braucht daher einen eigenen Editor, der sich im Optimalfall nicht vom Android-Standard-Widget unterscheidet. Dadurch wird gewährleistet, dass Benutzer intuitiver mit der Applikation interagieren können. Es ist sinnvoll, für einfachen Text die integrierte Tastatur, für Zahlen die gewöhnliche Nummerneingabe und für Daten die gewöhnliche Kalendereingabe zu verwenden. Lediglich die Bereichsauswahl für Zahlen stellt eine Eingabe dar, für die Android keinen Standardeditor anbieten kann. Dieser wird konzeptionell beibehalten, jedoch wird das Feld von Buttons aufgelockert und ab sofort als modaler Dialog dargestellt, um den Fokus des Benutzers entsprechend zu lenken. Die Buttons werden farblich hervorgehoben, damit der Benutzer ein besseres Feedback bekommt, was ausgewählt wurde [1, Pickers]. Zudem soll der verfügbare Platz somit besser genutzt werden [6].

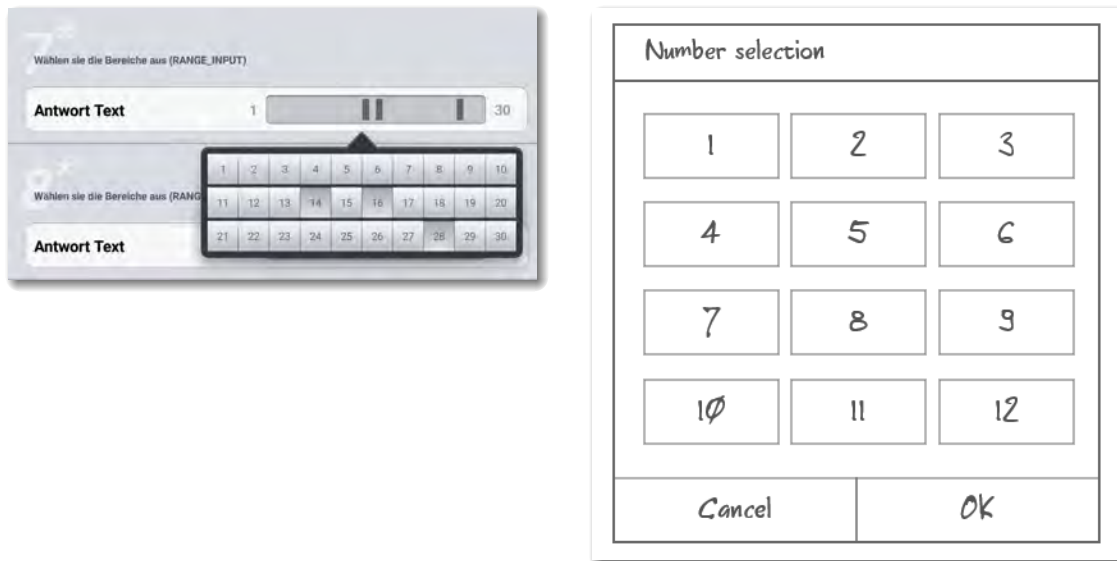


Abbildung 4.10: Bereichsauswahl im Ausgangszustand links und rechts das neue Konzept

Zwei-, Drei, und Mehrstellungsschalter

Die Zwei- und Dreistellungsschalter (siehe Abbildung 4.11) sollen lediglich einen neuen Look bekommen. Die grundsätzliche Logik der Oberfläche, also welche Stellungen die Schalter einnehmen können, kann beibehalten werden. Das neue Design orientiert sich dabei an den üblichen Android-Slidern und -Switches. [1, Sliders]

Am Mehrstellungsschalter ist keine Änderung geplant, da es für seine Funktionalität keine expliziten Vorgaben seitens des Material Design Guides gibt. Die von Google



Abbildung 4.11: Zwei- und Dreistellungsschalter im Ausgangszustand links und rechts das neue Konzept



Abbildung 4.12: Mehrstellungsschalter im Ausgangszustand

angebotenen Optionen erfüllen nur teilweise die angestrebte Funktionalität, da sie keine neutrale Ausgangsposition anbieten. Daher ist die eigens für diese Funktion entwickelte Komponente aus der bereits bestehenden Applikation die sinnvollste Option. [1, Sliders]

Navigation

In Abbildung 4.13 ist die ursprüngliche Navigation innerhalb des Fragebogens dargestellt. Diese hat nichts mit der aus der Startseite bekannten Navigation zu tun. Jedoch ist diese Form der Navigation mit Buttons am oberen Rand des Bildschirms nicht für Android vorgesehen. Aus diesem Grund soll hier eine andere Form der Navigation zwischen den einzelnen Seiten eines Fragebogens zum Einsatz kommen. Eine gute Möglichkeit bietet der *Floating Button*, der einen Button darstellt, welcher – wie der Name schon vermuten lässt – über die Oberfläche *fließt*. D.h., dass er sich beim Scrollen nur relativ zur Oberfläche unter sich bewegt, also sich relativ zum Bildschirm nicht bewegt. Üblicherweise wird er in der unteren rechten Ecke des Bildschirms positioniert. Hinzu kommt, dass die Validierungsfunktion erhalten bleiben soll, die an der ursprünglichen Navigation hängt. Es soll erst dann möglich sein, auf die nächste Seite zu gelangen, wenn alle obligatorischen Fragen beantwortet wurden. Im der alten Applikation wurde dann der Button aktiv geschaltet, was an einer Blaufärbung in Abbildung 4.13 zu erkennen ist.



Abbildung 4.13: Seitennavigation links vor und rechts nach dem Ausfüllen obligatorischer Fragen

4.3 Fragebogenelemente

Analog dazu soll der Floating Button aus Abbildung 4.14 erst sichtbar werden, sobald die Validierung positiv erfolgt ist. [1, Floating Action Button]

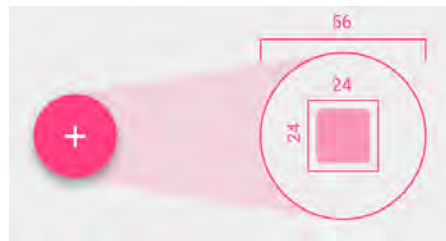


Abbildung 4.14: Floating Action Button [1, Floating Action Button]

5

Umsetzung und Implementierung

Dieses Kapitel befasst sich mit der Umsetzung der Konzepte aus Kapitel 4 und deren konkrete Implementierung für die Android-Plattform. Dabei wird exemplarisch Bezug auf Designelemente und Quellcode genommen, wie z.B. XML-Dateien für die Gestaltung des Layouts. Außerdem findet ein Abgleich mit den Anforderungen statt, und in welcher Hinsicht aus diesen gepaart mit den Konzepten schlussendlich die nachfolgende Implementierung resultierte. Dazu wird parallel zur Gliederung aus Kapitel 4 auf die Elemente eingegangen und diese jeweils anhand einer Visualisierung erläutert, die von der neu implementierten Applikation so realisiert werden. Die Screenshots stammen dabei im Allgemeinen von einem Tablet. Lediglich zu Anschauungszwecken kann an geeigneter Stelle auch ein Bild aus einer anderen Quelle mit einfließen, um bestimmte Effekte – wie z.B. Portabilität – besser präsentieren zu können.

5.1 Ressourcenmanagement

Das Ressourcenmanagement bezieht sich auf die Verwaltung der zusätzlichen Ressourcen unter Android. Darunter fallen Dimensionierungen, Farben, Texte, Styles und mehr, die in Form von XML-Dateien gespeichert und auch von dort abgerufen werden können. Das ist insofern relevant, dass einige Teile davon vom Design Guide vorgegeben werden, bzw. die folgenden Elemente der Implementierung damit arbeiten. Exemplarisch betrachten wir grob einige Teile davon – insbesondere die Farbe. Für die Farben wurden auch Tools (z.B. [7]) verwendet, mit denen eigene Farbpaletten zusammengestellt werden können.

Die Abbildung 5.2 zeigt einen aktuellen Ausschnitt aus der *color.xml*-Datei der Applikation inklusive Visualisierung durch eine Farbpalette. Zu sehen sind die für Android Lollipop typischen Farben, die man für das Material Design benötigt und vom entsprechenden Material-Theme verwendet werden. Themes und Styles werden dazu verwendet, allgemeine Designs festzulegen. So kann z.B. ein Style mit verschiedensten Parametern versehen und einem Button zugewiesen werden. Dadurch kann man Elemente einheitlich stylen, ohne dies für alle einzeln erledigen zu müssen [8]. Das bezieht sich auch auf Parameter wie Schriftgrößen, Paddings, Margins und spezifische Werte für unterschiedliche Bildschirmorientierungen, API-Level und Sprache. Letzteres kann für Mehrsprachigkeit ausgenutzt werden, indem man sprachbezogene XML-Dateien als Ressourcen verwendet.

Außerdem kann man damit zentralisiert und nicht-redundant Farbzuzuweisungen machen, die sich auf alle Bereiche der Applikation auswirken. So kann z.B. die Akzentfarbe als Referenz für andere Farbzuzuweisungen verwendet werden. Will man einen Button damit färben, bzw. Text oder bestimmte Highlights, dann weist man diese in der *color.xml* oder auch in der *styles.xml* zu. Daraus folgt, dass man nur noch an einer Stelle die Farbe ändern muss, um all diese Zuweisungen auch anzupassen. Allgemein kann so sehr einfach und schnell eine Anpassung der Farben vorgenommen werden. Ein Beispiel dazu ist in Abbildung 5.3 zu sehen.

DARK PRIMARY COLOR #455A64	PRIMARY TEXT #212121
PRIMARY COLOR #607D8B	SECONDARY TEXT #727272
LIGHT PRIMARY COLOR #CFD8DC	TEXT/ICONS #FFFFFF
ACCENT COLOR #009688	DIVIDER COLOR #B6B6B6

Abbildung 5.1: Farbpalette

```

1 <resources>
2   <color name="primary">#607D8B</color>
3   <color name="primary_dark">#455A64</color>
4   <color name="primary_light">#CFD8DC</color>
5   <color name="accent">#009587</color>
6   <color name="primary_text">#212121</color>
7   <color name="secondary_text">#727272</color>
8   <color name="icons">#FFFFFF</color>
9   <color name="divider">#B6B6B6</color>
10 </resources>

```

Listing 5.1: Ausschnitt aus *color.xml*-Datei

Abbildung 5.2: Visualisierung der Farbgebung durch XML-Ressourcen anhand eines Code-Ausschnitts aus der Applikation und einer Farbpalette

Dort sind auch Farben zu sehen, die auf Grund verschiedenste Kriterien nicht weiter in Frage kamen. So sind zu starke Rottöne für Aktionen nicht in Einklang mit dem Design Guide, der besagt, dass damit Negativassoziationen einhergehen. Dazu gehören z.B. die Buttons eines Dialogs, mit denen man Dialoge abbricht oder bestätigt. Aber auch die Farben einiger anderer Elemente sind subjektiv unangenehm. So sind Slider und Texte in einer offensiven Farbe unangebracht. Im Design Guide wird das allgemein schon so gehalten, dass eine Akzentfarbe nicht für allgemeine Texte genutzt, sondern lediglich zum Hervorheben von Links oder Schlüsselwörtern verwendet werden sollte. Hinzu kommen dann dem gegenübergestellt zu helle und kontrastarme Farben, die gerade für Text nicht gut geeignet sind, wenn man einen hellen Hintergrund hat. Letzten Endes

5 Umsetzung und Implementierung

fiel deshalb die Wahl auf einen Kompromiss von der Standard-Farbe für Android als Akzentfarbe und einem sachlichen aber ansprechenden Graublau-Ton als Primärfarbe. Diese Kombination stellt keinen Anspruch auf Optimalität, erfüllt aber auf jeden Fall die Anforderungen. Zu sehen ist ein Beispiel dazu in Abbildung 5.2 In Hinblick auf ältere Menschen ist es wichtig, Farben nicht zu sehr zu mischen [9]. Rot und gelb können von älteren Menschen besser unterschieden werden. Jedoch eignen sich diese Farben nicht [10]. Gelb bietet einen schlechten Kontrast auf weiß. Da blau aber einen guten Kontrast bietet, wurde die vorliegende Akzentfarbe ausgewählt. Beide Farben werden nur in Kontrast zu weiß eingesetzt, wodurch das Problem der Farbunterscheidung auch behandelt wird.

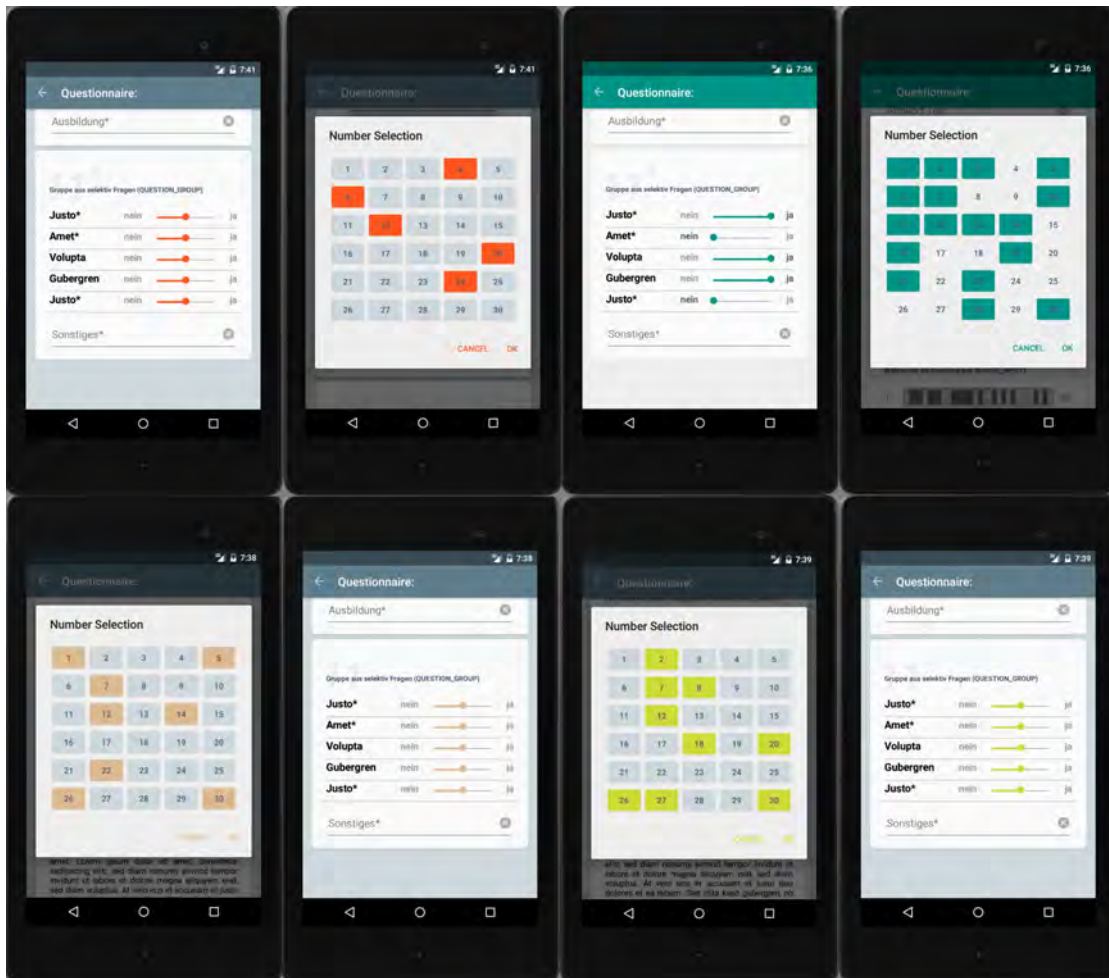


Abbildung 5.3: Verschiedene Farben – getestet auf einem Nexus 7 Emulator

5.2 Startseite

Nachdem die Farben und anderen Ressourcen für die Applikation klar sind, kommen nun einige Detailansichten bestehend aus Screenshot-Ausschnitten, die – sofern nicht anders erwähnt – von einem realen Nexus 7 stammen.

Toolbar

Die Toolbar im neuen Design wurde sehr ähnlich zum Konzept aus Abbildung 5.5 realisiert. Lediglich das Menu Icon entfällt, da es zumindest bisher nicht genügend Einstellungen und Aktionen gibt, die darin untergebracht werden können. Der typische Farbverlauf für Applikationen in Android Lollipop von dunkel nach hell wurde auch nachempfunden, sowie im Ansatz auch die perspektivische Erhöhung durch den Schattenwurf der Toolbar auf die nachfolgende, *darunter* liegende Ebene. Dasselbe tritt auch bei dem Auswahldialog auf, welcher sich auf der rechten Seite befindet (siehe Abbildung 5.4). An dieser Stelle ist ein Dropdown-Element integriert, der bei seiner Aktivierung den zu sehenden Auswahldialog öffnet. Der nachfolgende Inhalt in dieser Abbildung spielt auf die in diesem Kapitel noch folgende Fragebogenliste an, und das aus Abbildung 4.4 bekannte *Burger Icon* ist eine Referenz auf den Navigation Drawer, welcher nachfolgend betrachtet wird.

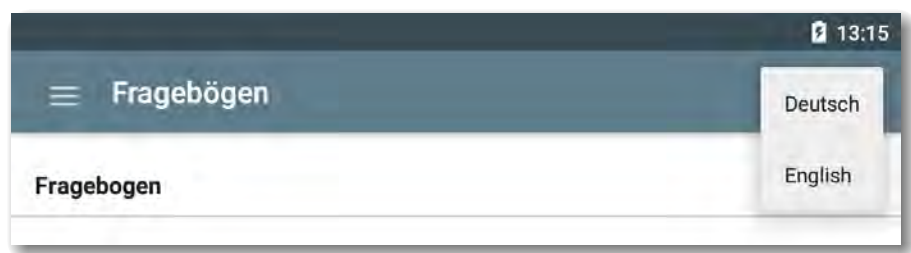


Abbildung 5.4: Toolbar mit geöffneter Sprachauswahl

Navigation

In Abbildung 5.5 ist nun statt dem Platzhalter *HEADER* aus Abbildung ?? das Branding der Applikation *Questionnaire* zu sehen. Beim Aktivieren des Burger Icons in der Toolbar wird diese Ansicht geöffnet. Dabei ist die aktuelle Seite grau hinterlegt, und die einzelnen Navigationsoptionen in Subkategorien unterteilt. Hinzu kommt, dass android-spezifische Icons verwendet werden, um den Wiedererkennungswert zu erhöhen. Sehr bekannt ist z.B. das Icon für die Einstellungen, welches durch ein Zahnrad repräsentiert wird. Zudem geht der Drawer über die ganze Bildschirmhöhe – abgesehen von der Navigationsleiste ganz unten am Bildschirm. Dabei sind sehr viele Optionen bereits vorgegeben. Allerdings werden dafür zunehmend unterstützende Bibliotheken und Funktionen bereitgestellt. Für die in dieser Arbeit entwickelte Applikation wurde die Android Design Support Library [11] verwendet, die einige Elemente des Material Designs beinhaltet und auch für ältere Betriebssystem-Versionen bereitstellt. Dabei unter anderem Support für den Navigation Drawer und den Floating Button.

Fragebogenliste

Der Material Design Guide gibt auch Auskunft über die Verwendung von Listen – insbesondere die Variante, die ausklappbare Elemente verwendet. Allerdings wird bis auf die Spezifikation des Designs und der Verwendung nichts bereitgestellt. Die durch das Android SDK bereitgestellte Variante dieser Liste ist noch veraltet und entspricht nicht dem Stand der Technik. Darum wurde die Fragebogenliste in Anlehnung an das originale Listenelement selbst implementiert (siehe Abbildung 5.6). Auf Icons zu Beginn der Listeneinträge wurde allerdings verzichtet, da es sich hier jeweils um denselben Typ von Eintrag handelt – nämlich Fragebögen. Das entsprechende Icon auf der rechten Seite zum auf- bzw. zuklappen wurde passend übernommen und auch das farbliche Hervorheben des Textes, wenn ein Listeneintrag ausgeklappt wird. Die Auswahl ist dabei so implementiert, dass immer nur ein Eintrag aufgeklappt sein kann. Wird ein anderer Eintrag ausgewählt, so schließt sich ein gegebenenfalls offener anderer Eintrag automatisch, bevor sich der neue öffnet. Dies soll gewährleisten, dass Benutzer nur die

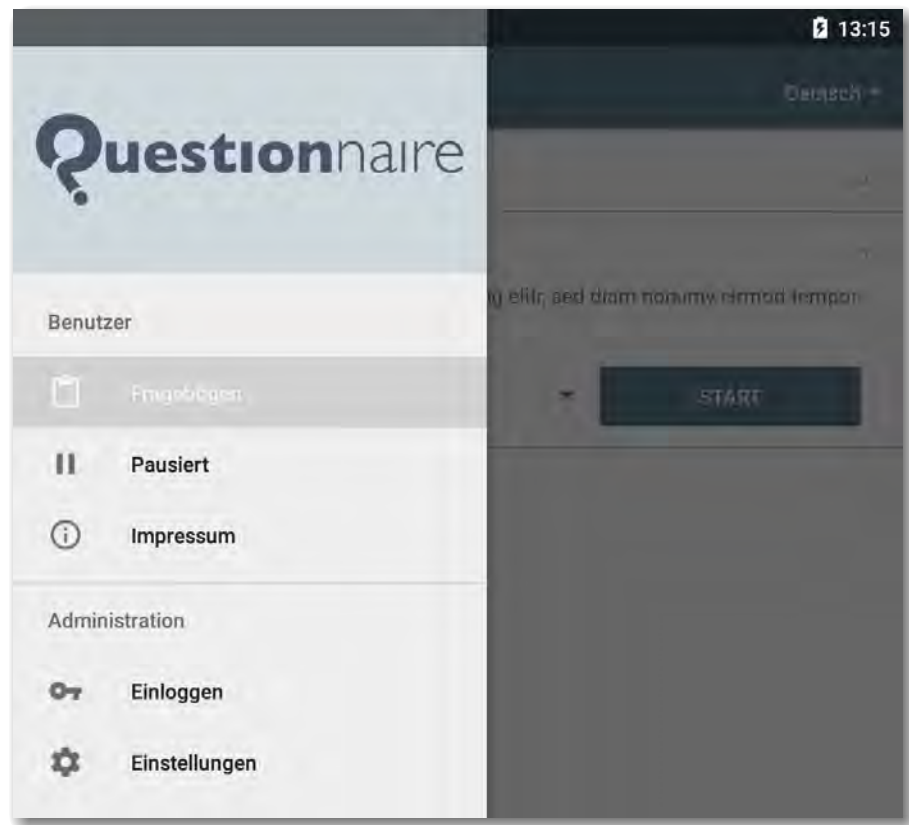


Abbildung 5.5: Navigation Drawer

Informationen sieht, die er benötigt. Zwischen dem Titel und den Bedienelementen für das Listenelement findet sich außerdem Platz für einen Text.

Die Sprache für den Fragebogen lässt sich über ein in die Oberfläche eingelassenen Dropdown-Button auswählen, dessen Funktion in Abbildung 5.6 zu sehen ist. In der ausgewählten Sprache kann dann der Fragebogen gestartet werden. Dazu dient der direkt angrenzende Button. Außerdem ist ein Ripple-Effekt integriert, der in Android Lollipop häufig Anwendung findet. Zu sehen ist diese Eigenschaft z.B. für den Listeneintrag in Abbildung 5.6. Aber auch andere Elemente besitzen diese Eigenschaft.

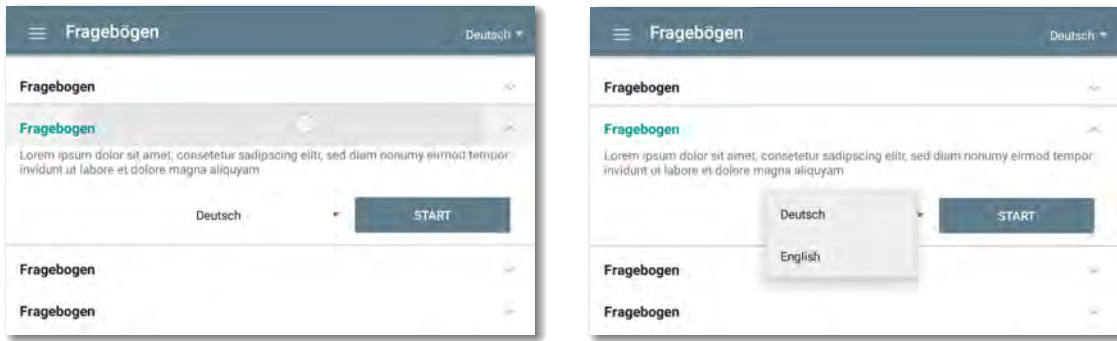


Abbildung 5.6: Fragebogenliste links mit aktivem Ripple-Effekt und rechts mit geöffneter Sprachauswahl

Impressum

Das neue Impressum in Abbildung 5.7 besteht aus einem Logo der Universität Ulm [12] und einigen Texten, gefärbt in der Primärfarbe der Applikation.



Abbildung 5.7: Impressum

Login

Klickt man in der Navigation auf *Einloggen*, so öffnet sich der Login-Dialog. Dieser soll laut Material Design Guide in Abschnitte mit bestimmten Maßen eingeteilt werden. Hauptsächlich sind das Titel, Inhalt und Aktionen. In diesem Fall besteht der Inhalt aus zwei Textfeldern mit sogenanntem Floating Label, welches man sogar in Aktion sehen kann. Tippt jemand ein solches Feld an, wird der Platzhalter-Text in dem Feld nach oben

5 Umsetzung und Implementierung

bewegt und in der zugewiesenen Farbe (beispielsweise die Akzentfarbe der Applikation) eingefärbt. Die Grundlinie des Feldes selbst wird ebenfalls in dieser Farbe gefärbt, bis sich der Fokus ändert. Abbildung 5.8 zeigt die Implementierung des Login-Dialogs. Das Grundkonzept für alle Dialoge ist identisch und genau vom offiziellen Design Guide spezifiziert. Dazu gehört z.B. das in der Farbgestaltung schon erwähnte Vermeiden von roter Schrift für die Buttons. *(Als Randnotiz soll hier vermerkt werden, dass diese Art von Button im Ruhezustand nicht als Buttons erkennbar sind. Diese sind in ihrem z-Wert nicht von ihrem Hintergrund verschieden. Allerdings haben sie sehr wohl eine Begrenzung. Sichtbar wird diese z.B., wenn man diese fokussiert.)*

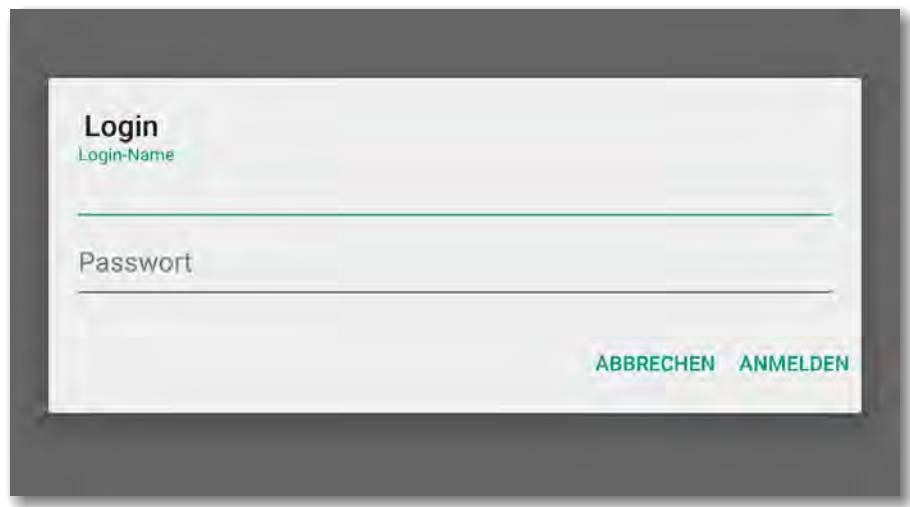


Abbildung 5.8: Login-Dialog

5.3 Fragebogenelemente

Im Allgemeinen musste eine neue Strukturierung und ein neues Design für Seite eines Fragebogens erstellt werden. Zu diesem Zweck wurden sogenannte *Cards* eingesetzt, die in die bestehende Listenstruktur eingebunden wurden. Sie dienen sozusagen als Wrapper für die einzelnen Elemente, welche dann gerendert und in die Liste eingesetzt werden. Der Styleguide erlaubt die Verwendung sogar ausdrücklich, da sie besagt: „Cards are a convenient means of displaying content composed of different elements.“

They're also well-suited for showcasing elements whose size or supported actions vary, like photos with captions of variable length.“ [1, Cards] Daraus folgt, dass man mit Cards gut verschiedene zusammengesetzte Elemente zur Schau zu stellen kann. Da es sich bei Fragebogenelementen um zusammengesetzte Elemente handelt, die teilweise in der Größe variieren, sollten Cards als Darstellungselemente für diesen Zweck verwendet werden.

Überschrift und Text

Die Überschrift und der Text bzw. Text mit optionaler Überschrift, sind nun wie in Abbildung 5.9 realisiert. Das erste Element darin stellt die reine Überschrift dar, das zweite den Text mit optionaler Überschrift. Die optionale Überschrift kann entfallen, dabei wird dann dieser Platz nicht beansprucht, sondern nur der reine Text angezeigt.

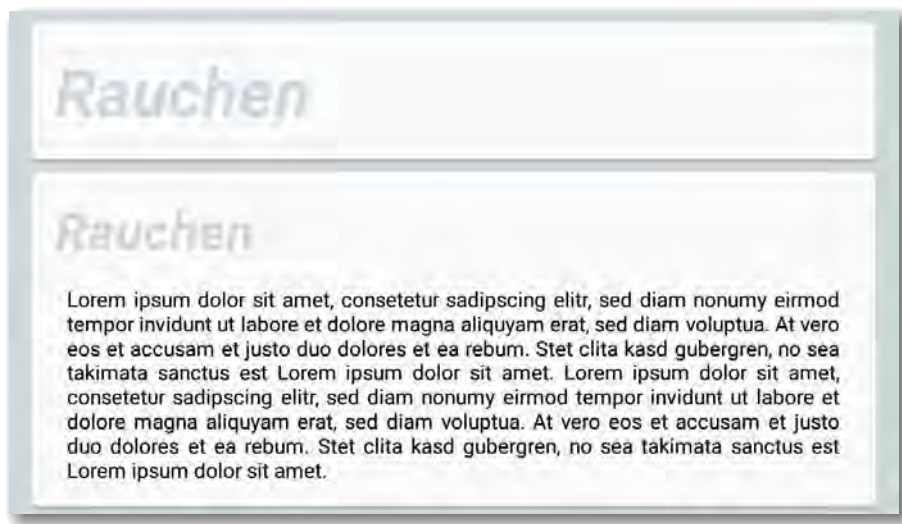


Abbildung 5.9: Text und Überschrift

Eingabefelder

Als Eingabefelder werden alle Felder bezeichnet, die eine Eingabe entgegen nehmen und anzeigen. In dieser Applikation fallen darunter Textfelder – beispielhaft dafür steht

5 Umsetzung und Implementierung

das Texteingabefeld aus Abbildung 5.10. Zusätzlich existiert ein Löschen-Button am Ende des Feldes, um die getätigte Eingabe wieder aus dem Eingabefeld zu löschen. Und zuletzt gibt es noch das Eingabefeld der Bereichsauswahl, welches in Abbildung 5.11 zu sehen ist.

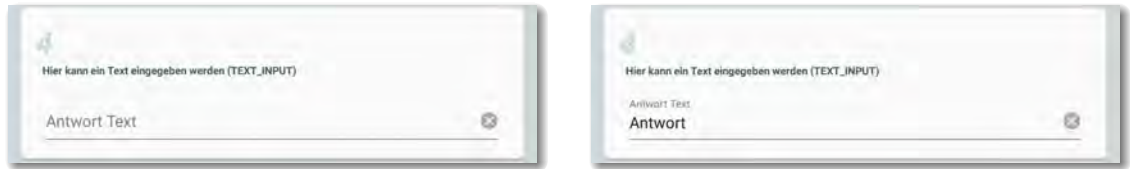


Abbildung 5.10: Texteingabefeld links ohne und rechts mit Eingabe

Letzteres ist an sich unverändert erhalten geblieben und wurde lediglich neu eingebettet. Die Funktion des bestehenden Elements ist gut und es lies sich nichts finden, das gegen diese Darstellung der Daten spricht. Dabei wird aus einem entsprechenden Editor eine Auswahl an Zahlen getroffen. Dieser zusammenhängende Zahlenbereich wird durch dieses Element repräsentiert und für jede ausgewählte Zahl wird an entsprechender Stelle ein Anteil dunkel verfärbt. So entstehen die eingefärbten Bereiche, die in der genannten Abbildung zu sehen sind.



Abbildung 5.11: Eingabefeld für die Bereichsauswahl links ohne und rechts mit Eingabe

Und auf die angesprochenen zugehörigen Editoren wird im nachfolgenden Kapitel eingegangen.

Editoren

Zu den Editoren gehören die Tastaturen, die verwendet werden, um Eingaben in die Eingabefelder zu tätigen. Im Wesentlichen gibt es die Tastaturen, die allerdings vom Nutzer selbst gestaltet werden, da die im System hinterlegten verwendet werden. Und diese kann man sich selbst aussuchen. Ein Beispiel, wie so etwas auf dem Testgerät

Nexus 7 aktuell aussieht, ist in Abbildung 5.12 zu sehen. Erstere Abbildung zeigt die Texteingabe, zweitere Abbildung zeigt die Zahleneingabe.

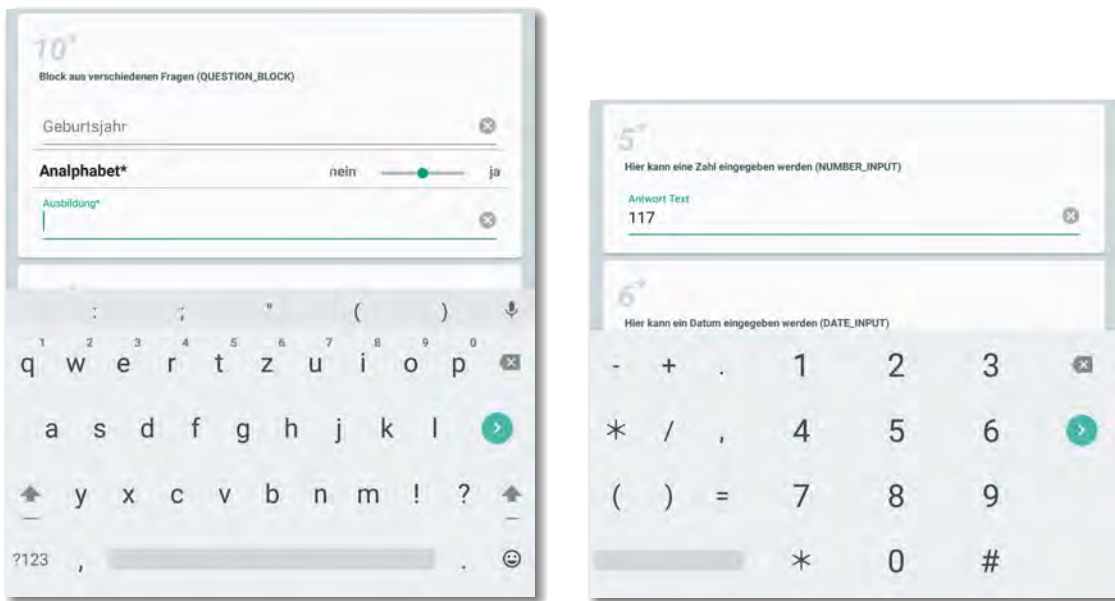


Abbildung 5.12: Tastaturen für Text- und Zahleneingabe

Ein weiteres Element zur Eingabe ist der Editor für das Datum. Allgemein gibt es hierfür bereits eine vorgefertigte Dialog-Version der Android-Plattform. Allerdings war hier der Wunsch, die Dialoge möglichst einheitlich und generisch zu gestalten, damit man diese gegebenenfalls leichter ändern und anpassen kann. In Abbildung 5.13 ist der Kalender zur Datumsauswahl in der Portrait-Ansicht zu sehen. Die Buttons am unteren Rand, die Highlight-Farbe des ausgewählten Tages und die Hintergrundfarbe des Kalenders stammen hierbei wiederum von der definierten Akzentfarbe der Applikation ab.

Derselbe Editor ist in Abbildung 5.14 in der Landscape-Ansicht zu sehen. Der Dialog wird bei diesem Vorgang geschlossen und mit neuen Attributen wieder geöffnet, damit er diese Ansicht erzielt. Intern wird dabei ein Kalender-Element verwendet, das von Android bereitgestellt wird und so auch in deren bereits vorgefertigtem Dialog vorzufinden wäre.

Als letztes bleibt noch der Editor zur Bereichsauswahl, welcher von der Grundidee beibehalten wurde. Allerdings musste dieser neu umgesetzt werden, um dem Material Design gerecht zu werden. Als erstes wieder die Portrait-Ansicht in Abbildung 5.15,

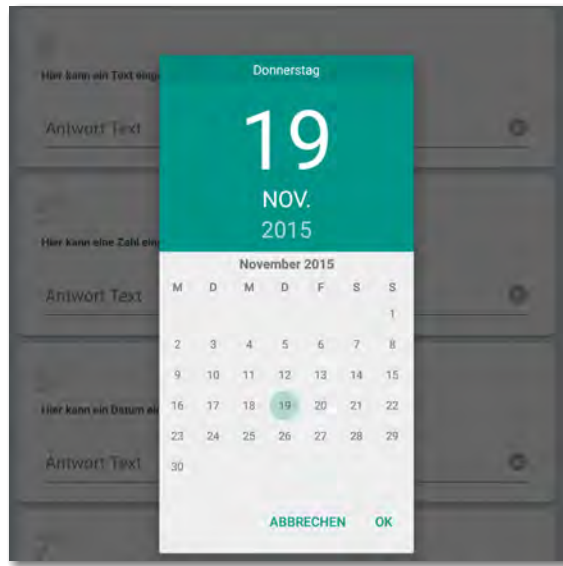


Abbildung 5.13: Kalender in Portrait-Ansicht

links vor der Auswahl, rechts nach der Auswahl einiger Zahlen aus dem gegebenen Zahlenbereich. Zu verbessern war allerdings die Flächennutzung der Buttons. Dazu zählt die Möglichkeit, präzise einen bestimmten Button anzuvisieren und anzuklicken, ohne dass man aus Versehen einen anderen aktiviert. Außerdem zählt dazu auch die Darstellung von größeren Zahlenbereichen, bei denen nicht auf konstanter Fläche immer kleinere Buttons erscheinen dürfen, da auch die erstgenannte Bedingung stark darunter leiden würde. Dazu wurde eine angemessene Größe der Buttons festgelegt und in einem Grid untergebracht, welcher dann in einem Dialog untergebracht wird. Letzteres kommt wiederum der Raumnutzung entgegen, da der Dialog mehr Platz auf dem Bildschirm einnimmt.

Besonders ist hier auch die Spaltenanzahl, die von der Geräteorientierung abhängig ist. So hat der Grid in Portrait-Ansicht hier fünf Spalten, in denen die Buttons angeordnet werden. Wohingegen Abbildung 5.16 die Landscape-Ansicht zeigt, bei der acht Spalten verwendet werden. Dies lässt sich einfach über ausgelagerte Parameter erzielen, die in XML-Ressourcendateien hinterlegt werden und abhängig von definierten Bedingungen ausgewählt werden. So gibt es z.B. die Möglichkeit, Dimensionierungen in eine XML-Datei abzulegen, die verwendet wird, wenn sich das Gerät im Landscape-Modus



Abbildung 5.14: Kalender in Landscape-Ansicht

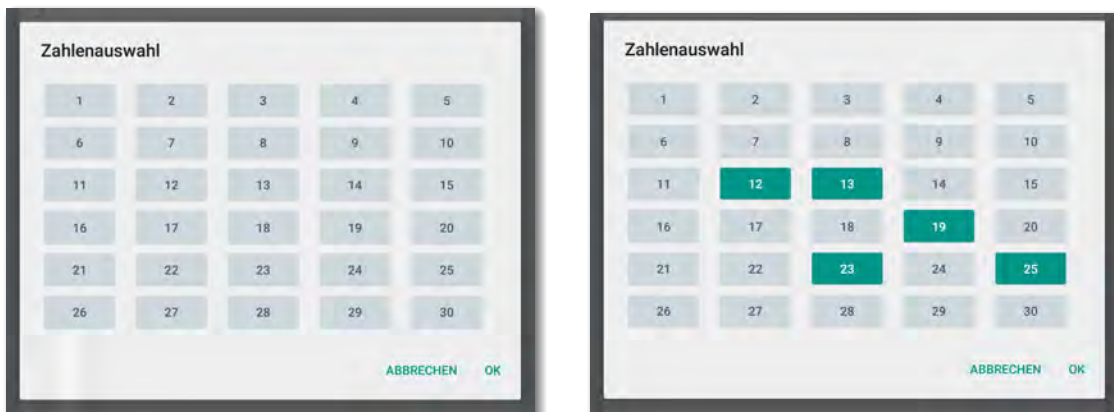


Abbildung 5.15: Editor für Bereichsauswahl in Portrait-Ansicht

befindet. Analog dazu definiert man dann auch einen Default-Wert, der dann für den Portrait-Modus verwendet wird.

Leider kann man in Hinsicht auf die Benutzerfreundlichkeit nicht gleichzeitig ein scrollfähiges Grid verwenden und dazu noch Wisch-Gesten zur Auswahl der Buttons. Denn eine vertikale Bewegung würde als Scrollen anstatt Auswählen interpretiert werden. Was bedeuten würde, dass man rein horizontale Bewegungen betrachten müsste. Da allerdings eine gewisse Abweichung in der Vertikalen ganz natürlich sein dürfte, müsste man diesen Umstand ebenfalls miteinbeziehen. Aus diesem Grund blieb es bei einfachen Buttons, die angeklickt werden können. Jedoch nicht zuletzt musste dafür noch das visuelle Feedback angepasst werden, da in der alten Version dieses Editors nur marginale Unterschiede zwischen einem aktivierten und einem nicht aktivierten Button

5 Umsetzung und Implementierung



Abbildung 5.16: Editor für Bereichsauswahl in Landscape-Ansicht

bestanden. In der neuen Version wird der Button in einer dezent dunkleren Farbe als der Hintergrund gefärbt, damit man die Begrenzungen des Buttons noch wahrnimmt. Zudem wird der aktivierte Button in der Akzentfarbe und der enthaltene Text in weiß gefärbt. So bleibt auch ein guter Kontrast erhalten.

Zwei-, Drei-, und Mehrstellungsschalter

Die neuen Zweistellungs- und Dreistellungsschalter basieren – wie schon die Vorgänger-variante – auf einem Android-Slider [1, Sliders]. Allerdings war in diesem Fall wichtig, dass das ganze besser sichtbar war und natürlich wiederum an das neu Design adaptiert wird. Abbildung 5.17 zeigt links die beiden Schalter vor und rechts nach einer Eingabe. An sich sind die Schalter identisch, bis auf die dahinter liegende Logik. Beim Zweistellungsschalter ist die neutrale Ausgangsposition nicht als Auswahloption gültig

und nach einer Eingabe nicht mehr erreichbar. Beim Dreistellungsschalter hingegen ist die neutrale Position wieder auswählbar und stellt eine valide Option dar.



Abbildung 5.17: Zwei- und Dreistellungsschalter links vor und rechts nach der Eingabe

Normalerweise wird bei diesen Slidern jedoch auch der Hintergrund bei einer Eingabe eingefärbt. Das könnte allerdings eine Wertung implizieren, wenn eine Eingabe getätigt wird. Außerdem würde sich damit die neutrale Position als Antwort von den anderen Möglichkeiten abheben. Bzw. durch das Färben des Hintergrunds würde der Auswahlzustand nicht eindeutig gekennzeichnet. Aus diesen Gründen folgte, dass nur der *Thumb*, also das bewegliche Teil des Sliders, in der Akzentfarbe gefärbt werden soll. Der Hintergrund wird etwas dicker gezeichnet, als bei einem üblichen Slider, damit man das Objekt besser sehen und vom restlichen Hintergrund unterscheiden kann. Außerdem wird der Text der Antwort bei einer Eingabe mit eingefärbt, um ein besseres Feedback zu geben, zumal der Hintergrund des Sliders nicht mehr eingefärbt wird und bis auf die Farbe des Thumbs selbst kein Feedback mehr bietet.

Der Mehrstellungsschalter wurde beibehalten (siehe Abbildung 5.18). Er verstößt nicht gegen Konventionen des Design Guides und passt in das übrige Design. Lediglich die Umgebung wurde – wie bei allen Elementen – angepasst, und das Element etwas gestreckt.

Navigation

Zu guter Letzt kommt noch der Floating Action Button mit einem System-Icon [1, Icons] von Android. Dieser ist beim Start eines Fragebogens nicht sofort sichtbar. Wird der Fragebogen bearbeitet und hat ein Benutzer alle als obligatorisch markierten Fragen

5 Umsetzung und Implementierung

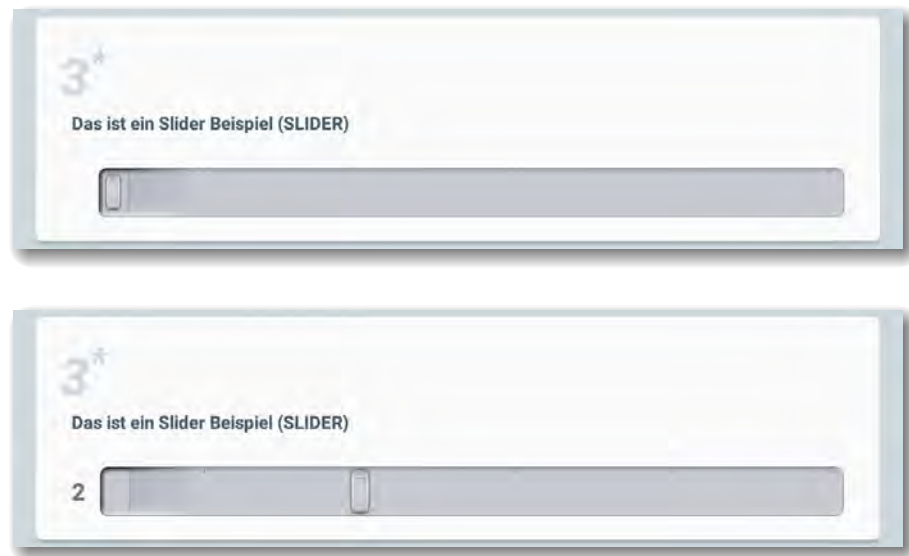


Abbildung 5.18: Mehrstellungsschalter oben mit und unten ohne Eingabe

beantwortet, wird der Button durch eine interne Validierungsfunktion freigeschaltet und schließlich angezeigt. Dadurch kann dieser aktiviert werden, um auf die nächste Seite des Fragebogens zu wechseln (siehe Abbildung 5.19).

Der Vorteil dieses Buttons ist, dass er – anders als eine Button-Bar, also eine Leiste mit Buttons – nicht viel Platz benötigt, aber dennoch immer verfügbar bleibt. Wie der Name schon andeutet, *fließt* der Button auf dem Layout. D.h., wenn man sich ganz unten am Ende der Fragebogenseite befindet und wieder nach oben scrollt, um eine Antwort auf eine Frage am Anfang der Seite zu ändern, so bewegt sich der Button mit. Das bringt dem Benutzer einen Vorteil in der Bedienbarkeit, da er nicht extra wieder nach unten scrollen muss, sondern der Button direkt verfügbar bleibt. Zudem ist das

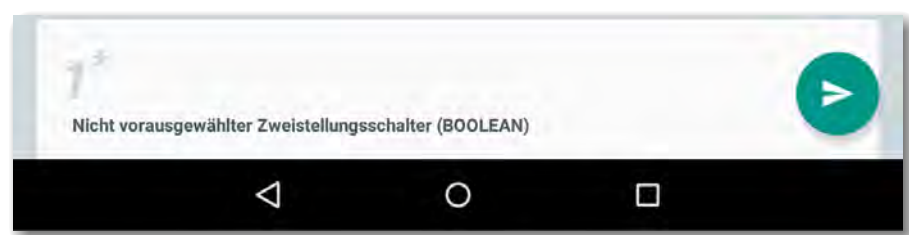


Abbildung 5.19: Screenshot mit aktivem Floating Action Button

5.3 Fragebogenelemente

Icon allgemein bekannt und intuitiv. Dadurch wird auch Mehrsprachigkeit unterstützt, ohne Text verwenden zu müssen. Außerdem wird dadurch eine unerwünschte Button-Bar umgangen und dank der Support Library [11] ist die Implementierung nicht weiter schwierig.

6

Anforderungsabgleich und Diskussion

Nachfolgend wird auf einige relevante Punkte eingegangen, die aus den Anforderungen bekannt sind. Zum Teil werden diese in größere Abschnitte zusammengefasst. Insgesamt soll nun betrachtet werden, wie sich die Anforderungen an die Applikation selbst und die Anforderungen des Design Guides im Allgemeinen miteinander vereinbaren lässt. Gegebenenfalls wird bei einer Diskrepanz auch erläutert, worin diese besteht. Außerdem wird ein Lösungsansatz vorgestellt und bewertet. Zuletzt werden dann noch Aspekte angesprochen und erläutert, die keine korrespondierende Anforderung innerhalb der Applikation haben, jedoch Einfluss auf diese ausüben.

Nicht-funktionale Anforderungen

Es folgen die Diskussion und der grobe Abgleich zu den nicht-funktionalen Anforderungen.

DN1 Aussehen und Interaktion: Ziel war es, das Aussehen und die Interaktion für Android Lollipop als Vorgabe zu nehmen, und damit das Design im Sinne des Material Designs zu gestalten. Allgemein stellte dies kein größeres Problem dar. Nur vereinzelt, gerade bei Details ist es unklar, wie diese konkret umgesetzt werden sollen. Seitens des Guides ist nur vorgegeben, wie es aussehen und was dabei beachtet werden soll. Allerdings weiß man, worauf es den Entwicklern dieser Richtlinien ankommt, und man versucht, sich dem möglichst anzunähern. Je nachdem, wie viel Zeit man bereit ist, zu investieren, bekommt man zumindest optisch, was die die Vorgabe verlangt. Oft ist man jedoch auf sich gestellt und muss gegebenenfalls auf Bibliotheken Dritter zurückgreifen. Diese oft privater entwickelt und geben entsprechend keine Garantie auf Vollständigkeit, Kompatibilität und Aktualität. Zuletzt bleibt zu erwähnen, dass das Material Design sich durch seine Schlichtheit gut für Fragebogen-Anwendung eignet [13].

DN2 Benutzbarkeit: Tatsächlich ergibt es Sinn, sich an die vorgegebenen Richtlinien zu halten. Denn diese unterstützen einen i.d.R. beim Einhalten bestimmter Anforderungen wie der Benutzbarkeit. Man sollte nur davon abweichen, wenn es zwingend nötig ist, sollte das dann aber auch bewusst und konsequent tun, um den Benutzer nicht zu verwirren. Was die Bedienbarkeit angeht, so ist man dabei schon fast auf der sicheren Seite, wenn man sich an den Design Guide hält. Natürlich muss man sich dennoch selbst Gedanken machen, wie man bestimmte Funktionen und Elemente sinnvoll kombinieren könnte oder sollte. Da sehr vielen Leuten das Betriebssystem Android bereits bekannt ist, ergibt es durchaus Sinn, ähnliche Aufgaben einheitlich zu gestalten. So wäre es unsinnig, eine eigene Tastatur zu implementieren, es sei denn, man hat spezielle Ansprüche an eine solche, die vom Standard nicht erfüllt werden. Allerdings hinkt die Unterstützung auf irgendeine Art beeinträchtigter Menschen seitens des Design Guides hinterher, was auch anderweitig als in dieser Arbeit festgestellt wurde [14].

DN3 Gestaltung: Bei der farblichen Gestaltung der Applikation wurde darauf geachtet, dass keine schlechten Kontraste oder ansonsten ungünstige Farbkombinationen verwendet wurden. Farben, die eine Negativassoziation verursachen können und auch speziell Rot-Grün-Kontraste wurden vermieden. Damit sollten verbreitete Einschränkungen bei Benutzern beachtet werden, bzw. tendenzielle Assoziationen, eventuell auch positiver Art, reduziert werden. Damit ist gemeint, dass das Ergebnis durch die Verwendung bestimmter Gestaltungsmittel in keine Richtung (positiv oder negativ) verfälscht werden soll. Der Design Guide hier nur begrenzt Unterstützung, da nur marginal auf solche Dinge eingegangen wird. Jedoch ist zu beachten, dass dieser allgemein gehalten ist und auch ermahnt – wenn auch ohne wirklichen Lösungsansatz – dass man sich Gedanken machen muss und die Zugänglichkeit auch für andere Benutzergruppen ermöglichen sollte [1, Usability]. Eine Möglichkeit, die sich im Nachhinein ergab, könnte eine Support-Bibliothek von Android sein (siehe Kapitel 7). Hinzu kommt, dass die für diese Anwendung ausgewählte Primärfarbe sogar in anderen Quellen verwendet wurde [15]. Dabei handelt es sich sogar um Design für ältere Menschen.

DN4 Betrieb und Performance: Die Ansprüche an die Performance und den Betrieb aus den Anforderungen decken sich mit denen des Design Guides. Daher war es problemlos möglich, diese miteinander zu vereinbaren. Schön zu sehen beispielsweise anhand der Maßeinheit der dichteunabhängiger Pixel, welche dazu dienen, dass dasselbe Layout gemessen in dieser Einheit auf anderen Geräten zu einer optimierten Anzeige führt. Wie bereits aus Kapitel 2 bekannt, orientieren sich die Entwickler des Material Designs an der Realität. Das bringt den Vorteil mit sich, dass durch subtiles Feedback von aktivierten Elementen ein unschöner Ladebalken nicht nötig ist. Für den Benutzer ist ersichtlich, dass die Anwendung auf seine Aktionen reagiert. Und bedingt durch die sanften Effekte des Material Designs fallen Unterschiede in der Performance weniger stark ins Gewicht. Beispielsweise dauert der Ladevorgang einer Fragebogenseite abhängig von der Menge der enthaltenen Elemente einen Augenblick länger. Hierbei hilft das Touch-Feedback des Startbuttons, dass einem Benutzer klar ist, dass die Anwendung seine Aktion entgegengenommen hat und nun arbeitet.

DN5 Wartbarkeit und Portierbarkeit: Es kommt der Wartbarkeit und Portierbarkeit natürlich entgegen, wenn man übersichtlich, modular und mit möglichst wenig fremden Bibliotheken arbeitet. Auch aus diesem Grund wurden einige Elemente selbst implementiert. Bibliotheken Dritter sind oft nur kurzfristig weniger Arbeit. Langfristig könnte man Änderungen vornehmen wollen und feststellen, dass diese nicht möglich sind. Daher sollte man Bibliotheken möglichst vermeiden, insbesondere die Dritter. Bibliotheken, die direkt von Google veröffentlicht werden, erhalten eher Support und Abwärtskompatibilität, weshalb diese in Ordnung sind. Allerdings sollte man sich dennoch Gedanken machen, ob sie langfristig den gesparten Aufwand wert sind, und man nicht einer selbst implementierten Variante bessere Optionen hat.

Funktionale Anforderungen

Es folgen die Diskussion und der Abgleich der funktionalen Anforderungen.

DF1 Mehrsprachigkeit: Was die Unterstützung von Mehrsprachigkeit angeht, so unterstützt einen der Design Guide durchaus. Er bietet Denkansätze, was bei der Gestaltung mehrsprachiger Oberflächen zu beachten ist. So gibt es Länder, in denen die Leserichtung umgekehrt ist. Zusammen mit dem Android-Betriebssystem bietet sich eine gute Grundlage z.B. mithilfe von XML Mehrsprachigkeit zu implementieren. Allerdings wird wiederum der Standardfall für mobile Anwendungen angesprochen, die einheitlich Sprachen verwendet. In Kontext dieser Arbeit sollten zwei verschiedene Sprachselektionen ermöglicht werden. Mit internen Mitteln ist dem nicht mehr allein nachzukommen, daher musste eine eigene Lösung entworfen werden.

DF2 Oberflächenelemente: Wenn man sich für Fragebogenelemente den Design Guide ansieht, so stellt man fest, dass gewöhnliche Elemente bereits definiert sind. Unter anderem sind das bewährte Elemente, wie der Navigation Drawer zur Navigation, oder Text- und Eingabefelder. Auch im weiteren Sinne sind teils Angaben vorhanden, wie etwas zu behandeln ist. Dazu gehört z.B. bei der Typografie, wie

breit ein Text gemessen in Zeichen sein sollte, damit dieser noch gut zu lesen ist. Oder auch für Dialoge, dass man die Textfarbe der Buttons dort nicht rot wählen sollte. Wenn etwas nicht vorhanden ist, ist einem im größeren Kontext dennoch klar, wie ein neues Element aussehen könnte, damit es sich in das Konzept des Material Designs nahtlos einfügt. So geschehen z.B. beim Editor der Bereichsauswahl, der einen Dialog darstellt, den es in dieser Form nicht im Design Guide gibt.

7

Zusammenfassung und Ausblick

Dieses Kapitel führt die gesammelten Erkenntnisse und Ergebnisse zusammen und gibt einen darauf basierenden Ausblick wieder. Der Ausblick beinhaltet unter anderem Ideen und Gedanken für weiterführende Themen.

7.1 Zusammenfassung

Eine unumgängliche Aufgabenstellung im Zeithalter der mobilen Geräte ist es, bestehende Anwendungen aktuell zu halten. Deren Design zusammen mit ihren Anforderungen soll dabei in Einklang mit dem offiziellen Design der zugehörigen Plattform weiterentwickelt werden. Da es sich unter Umständen um kontroverse Aspekte handelt oder man unterschiedliche Standpunkte des Designs vertritt, kann dieses Unterfangen schwierig sein. Man muss Kompromisse eingehen, die eine der beiden Seiten mehr oder weni-

ger stark beeinträchtigen. Zum einen will man seine Anforderungen nicht reduzieren. Man verfolgt ein konkretes Ziel und hat sich seine Gedanken dabei gemacht, diese Anforderungen aufzustellen. Zum anderen will man aber auch, dass sich die eigene Anwendung gut in die zugehörige Plattform eingliedert. Dazu sollte man den Benutzer in der Anwendung leiten und eventuell unter Eingeständnis von Kompromissen neue Kontrollelemente entwickeln. Hier geschah das konkret am Beispiel der Android-Plattform und ihren Material Design Guidelines. Was die Idee hinter diesem Designansatz ist, ist aus Kapitel 2 bekannt. Um aber nun auf die ursprüngliche Frage aus Kapitel 1 zurück zu kommen. Ja, es ist durchaus sinnvoll und praktikabel, auf den Design Guide zurückzugreifen, wenn man eine bereits bestehende Applikation neu gestalten möchte. Allerdings geht es dabei wohl weniger darum, diesen exakt einzuhalten. Viel wichtiger ist es, dass man den eigentlichen Gedanken hinter dem ganzen Design verstanden hat. Ist man an diesem Punkt angelangt, so kann man sich auch dafür entscheiden, vom Design Guide leicht abzuweichen. Jedoch sollte man sich dessen bewusst werden und Android-eigene Komponenten zwecks Kompatibilität verwenden. Das soll heißen, dass jemand, der weiß, was der Zweck einer Vorgehensweise ist, die der Design Guide einem nahelegt, bewusst einen Stilbruch wagen oder aber im Rahmen des Design Guides eine Alternative entwickeln kann. Man ist keineswegs so sehr an die Richtlinien gebunden, sollte sich bei den schnellen Release-Zyklen aber seiner eigenen Prioritäten bewusst werden. Vielmehr lebt dieser auch davon, dass Innovation betrieben wird. Lediglich die Botschaft, die damit vermittelt werden soll, muss man sich im Hinterkopf bewahren.

Deswegen ist der Lösungsansatz, man solle besser nicht zu jedem Minor Release das Konzept der eigenen Applikation komplett überdenken und stattdessen auf Major Releases des Betriebssystems hinarbeiten, durchaus erstrebenswert. Erst wenn man sich näher mit dem Thema des Designs befasst und dieses versteht, kann man ein aufeinander abgestimmtes Gesamtkonzept entwickeln, welches man daraufhin in die Tat umsetzen kann. Ich will meine Aussage nur insofern korrigieren, dass nicht einmal jeder Major Release wichtig oder auch nicht jeder Minor Release unwichtig sein muss. Und zuletzt gilt es zu bedenken, dass die Entwicklungen ungeheuer schnell von Statten gehen. Allein Im Laufe dieser Arbeit gab es unzählige Neuerungen, neue Support-Bibliotheken

und API-Änderungen. In der Zwischenzeit wurde sogar das nächste Major-Release Android 6.0 (Marshmallow) veröffentlicht.

7.2 Ausblick

Ungeachtet des Vorgehens bei einer Um- oder Neugestaltung einer Anwendung, gibt es noch Aspekte, die untersucht oder beachtet werden könnten oder sollten.

Im Einzelnen lässt sich sagen, dass z.B. für die Farbgestaltung noch mehrere Optionen offen sind, die ausprobiert werden könnten. Zum einen dürften den meisten die Bilder bekannt sein, die verwendet werden, um auf Rot-Grün-Schwächen zu testen [16]. Man sieht farbige Punkte und es wird eine Zahl dargestellt. Je nachdem ob man betroffen ist, sieht man eine andere Zahl. Dieses System könnte man mit der neuen Support-Bibliothek, die im vorangegangenen Kapitel 6 erwähnt wurde, testen. Dort ist es möglich, dass man aus einem Bild eine Farbpalette extrahiert und verwendet. So könnte man dynamisch reagieren und vorab z.B. einmalig pro Patient – sofern man sinnvollerweise bei so einem System die Patienten digital registriert und die gewonnenen in einer Datenbank Daten speichert – Konfigurationstests ausführen. Der Patient bekommt Bilder gezeigt mit unterschiedlichen Kontrasten, Farbkombinationen u.v.m., worauf dann eine speziell für diesen Patienten zugeschnittene Farbpalette generiert und verwendet wird. Alternativ könnte man auch unterschiedlich feine Abstufungen machen und Leute in Gruppen einteilen oder auch die Methode vereinfachen.

In einem größeren Rahmen betrachtet, könnte man mit den Farben auch andere Informationen verbinden. So könnte man ein vollständiges Konfigurationsprofil für einen Anwender erstellen, inklusive Sprachauswahl, Textgröße und vieles mehr. Loggt sich dann ein Administrator ein, so wird sein Profil geladen. Gibt er dann einen Patienten an, der ein Profil besitzt, wird dessen Profil geladen. Dabei würde man dann die verschiedenen Bereiche, in denen die jeweiligen Nutzer aktiv sind, entsprechend ihrer Profile gestalten. Dass es dafür Bedarf gibt, ist absehbar [17, 18].

Auch psychologisch fundierte neue Kontrollelemente könnten untersucht werden. Eine Beispiel wäre die Auswahl von Zahlenbereichen, die man z.B. im Rahmen von Multi-

7 Zusammenfassung und Ausblick

Touch weiterentwickeln könnte. Die Schwierigkeit dabei ist, dass man eine intuitive Bedienung erreichen muss, die gleichzeitig auch funktional ist. Dazu wäre es auch sinnvoll, die neuen Kontrollelemente in einer Studie mit Probanden auf ihre Qualität zu überprüfen (z.B. [19]).

Nicht zuletzt gibt es auch die Möglichkeit, andere Systeme zu betrachten. Das Android-Betriebssystem kommt an sehr vielen Stellen zum Einsatz. So gibt es neben Smartphones und Tablets auch noch Android für Smart TVs und Smartmatches. Diese verwenden andere Displaygrößen- und Formen. Dort benötigt man wahrscheinlich auch andere Kontroll- oder Navigationselemente. Wieder am Beispiel der Bereichsauswahl, könnte man den Ansatz verfolgen, die maximal mögliche Anzahl an Buttons pro Zeile genau zu berechnen. Dazu könnte man sich nach der *Minimum Target Size* richten, zu der es Paper gibt und die auch im Material Design Guide erwähnt wird [20].

Auch die Ausweitung der miteinbezogenen Beeinträchtigung in das Design könnte ein interessanter Aspekt sein, wie bereits anderweitig angerissen wird [21]. Also dass man sich nicht nur auf die großen Gruppen beschränkt, wie Rot-Grün-Blinde oder ältere Menschen. Man könnte sich intensiver damit befassen, was es für spezielle Ansprüche geben könnte und wie man auf diese eingehen kann.

Und zuletzt kann man auch noch das Material Design außerhalb von Android betrachten. Das kurz erwähnte Web-Interface, das dazu dient, Fragebögen erstellen zu können, könnte man auch in das Material Design überführen, damit es einheitlich aussieht. Bzw. man kann versuchen, dass es ein Kompromiss aus verschiedenen Plattformen bildet, da dieselbe Anwendung auch auf anderen Plattformen realisiert werden könnte.

Literaturverzeichnis

- [1] Google: (Material Design Guidelines) <https://www.google.com/design/spec/material-design/introduction.html>, Zugriffsdatum: 21.09.2015, 11:36 Uhr.
- [2] Reisser, A.: Technische Konzeption und Realisierung einer dynamisch generierten Anwendung für prozess-orientierte Fragebögen am Beispiel der mobilen Android Plattform. Diplomarbeit, University of Ulm (2014)
- [3] Blendinger, K.: Tablet-Applications for the Elderly: Specific Usability Guidelines. Master's thesis, University of Ulm (2015)
- [4] Randolph, S.A.: Color Vision Deficiency. Workplace health & safety (2013) 280
- [5] Yang, S., Ro, Y.M.: Visual contents adaptation for color vision deficiency. In: 2003 International Conference on Image Processing - ICIP 2003. (2003) I–453–6 vol.1
- [6] Nilsson, E.G.: Design patterns for user interface for mobile applications. Advances in Engineering Software (2009) 1318 – 1328 Designing, modelling and implementing interactive systems.
- [7] materialup: (Material Palette) <https://www.materialpalette.com/>, Zugriffsdatum: 21.09.2015, 13:49 Uhr.
- [8] Google: (Styles and Themes) <http://developer.android.com/intl/vi/guide/topics/ui/themes.html>, Zugriffsdatum: 06.12.2015, 13:33 Uhr.
- [9] Wijk, H., Berg, S., Sivik, L., Steen, B.: Color discrimination, color naming and color preferences in 80-year olds. Aging (Milan, Italy) (1999) 176—185

- [10] Jenny, B., Kelso, N.: Color Design for the Color Vision Impaired. Cartographic Perspectives (2007)
- [11] Google: (Design Support Library) <http://developer.android.com/intl/vi/tools/support-library/features.html#design>, Zugriffsdatum: 21.09.2015, 13:25 Uhr.
- [12] Universität Ulm: (Uni Ulm Logo) https://de.wikipedia.org/wiki/Datei:Uni_ulm_logo.svg, Zugriffsdatum: 05.12.2015, 21:56 Uhr.
- [13] Väättäjä, H., Roto, V.: Mobile Questionnaires for User Experience Evaluation. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems. CHI EA '10, New York, NY, USA, ACM (2010) 3361–3366
- [14] Park, K., Goh, T., So, H.J.: Toward Accessible Mobile Application Design: Developing Mobile Application Accessibility Guidelines for People with Visual Impairment. In: Proceedings of HCI Korea. HCIK '15, South Korea, Hanbit Media, Inc. (2014) 31–38
- [15] Devos, P., Jou, A.M., Waele, G.D., Petrovic, M.: Design for personalized mobile health applications for enhanced older people participation. European Geriatric Medicine (2015) 593 – 597
- [16] Wong, B.: Points of view: Color blindness. Nature Methods (2011) 441–441
- [17] Frohner, M., Urbauer, P., Forjan, M., Pohn, B., Gerbovics, F., Sauermann, S., Mense, A.: Development of an Android App in Compliance with the Continua Health Alliance Design Guidelines for Medical Device Connectivity in mHealth. Biomedical Engineering/Biomedizinische Technik (2012) 997–999
- [18] Peischl, B., Ferk, M., Holzinger, A.: Integrating user-centred design in an early stage of mobile medical application prototyping: A case study on data acquisition in health organisations. In: 2013 International Conference on e-Business (ICE-B). (2013) 1–11

- [19] Wich, M., Kramer, T.: Enhanced Human-Computer Interaction for Business Applications on Mobile Devices: A Design-Oriented Development of a Usability Evaluation Questionnaire. In: 2015 48th Hawaii International Conference on System Sciences (HICSS). (2015) 472–481
- [20] Leitao, R., Silva, P.A.: Target and Spacing Sizes for Smartphone User Interfaces for Older Adults: Design Patterns Based on an Evaluation with Users. In: 19th Conference on Pattern Languages of Programs. (2012) 1–16
- [21] Holzinger, A., Sammer, P., Hofmann-Wellenhof, R.: Mobile Computing in Medicine: Designing Mobile Questionnaires for Elderly and Partially Sighted People. In Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A., eds.: Computers Helping People with Special Needs. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 732–739
- [22] Google: (Android Developers) <http://developer.android.com/guide/index.html>, Zugriffsdatum: 21.09.2015, 13:25 Uhr.
- [23] Friesen, J., Smith, D.: Android Recipes: A Problem-Solution Approach for Android 5.0. Apress (2015)
- [24] Becker, A., Pant, M.: Android 5 - Programmieren für Smartphones und Tablets. dpunkt.verlag GmbH (2015)
- [25] Schobel, J., Schickler, M., Pryss, R., Reichert, M.: Process-Driven Data Collection with Smart Mobile Devices. In: 10th International Conference on Web Information Systems and Technologies (Revised Selected Papers). Number 226 in LNBI. Springer (2015) 347–362
- [26] Schobel, J., Schickler, M., Pryss, R., Maier, F., Reichert, M.: Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned. In: 10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps. (2014) 371–382
- [27] Scherle, S.: Konzeption und Evaluierung einer domänenspezifischen Modellierungsumgebung für prozessorientierte Fragebögen. Diplomarbeit, University of Ulm (2014)

- [28] Deuter, D.: A Generic Approach for Developing and Executing Electronic Questionnaires on the iOS Platform. Master's thesis, University of Ulm (2014)
- [29] Kar, S.: Usability-Based User-Centered Design of Android Applications. International Journal of Scientific Research and Management (IJSRM) (2014) 1367–1371
- [30] Hsiao, C.Y., Liu, Y.J., Wang, M.J.: Usability Evaluation of the Touch Screen User Interface Design. In Yamamoto, S., ed.: Human Interface and the Management of Information. Information and Interaction Design. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 48–54
- [31] Buskirk, T.D., Andres, C.: Smart Surveys for Smart Phones: Exploring Various Approaches for Conducting Online Mobile Surveys via Smartphones. Survey Practice (2013)
- [32] Rachuri, K.K., Mascolo, C.: Smart phone based systems for social psychological research: Challenges and design guidelines. In: Proceedings of the 3rd ACM workshop on Wireless of the students, by the students, for the students, ACM (2011) 21–24

Abbildungsverzeichnis

2.1	3D-Raum des Material Designs [1, Introduction]	7
2.2	Links korrekt mit 1dp Dicke und rechts falsch mit mehr als 1dp Dicke [1, Environment]	8
2.3	Schattenwurf, von links nach rechts: mit Schlüssel-Licht, mit Umgebungslicht, kombiniert [1, Elevation and Shadows]	9
4.1	Vereinfachtes Ablaufdiagramm der mobilen Applikation	20
4.2	Splashscreen im Ausgangszustand	21
4.3	Toolbar im Ausgangszustand	22
4.4	Toolbar-Konzept; Nav Icon oder auch Burger Icon (1) und Menu Icon (2)	22
4.5	Startseite links im Ausgangszustand und rechts das neue Konzept	23
4.6	Fragebogenliste links im Ausgangszustand und rechts das neue Konzept	24
4.7	Login-Dialog links im Ausgangszustand und rechts das neue Konzept	25
4.8	Überschrift und Text im Ausgangszustand	26
4.9	Eingabefelder im Ausgangszustand	26
4.10	Bereichsauswahl im Ausgangszustand links und rechts das neue Konzept	27
4.11	Zwei- und Dreistellungsschalter im Ausgangszustand links und rechts das neue Konzept	27
4.12	Mehrstellungsschalter im Ausgangszustand	28
4.13	Seitennavigation links vor und rechts nach dem Ausfüllen obligatorischer Fragen	28
4.14	Floating Action Button [1, Floating Action Button]	29

Abbildungsverzeichnis

5.1	Farbpalette	33
5.2	Visualisierung der Farbgebung durch XML-Ressourcen anhand eines Code-Ausschnitts aus der Applikation und einer Farbpalette	33
5.3	Verschiedene Farben – getestet auf einem Nexus 7 Emulator	35
5.4	Toolbar mit geöffneter Sprachauswahl	36
5.5	Navigation Drawer	38
5.6	Fragebogenliste links mit aktivem Ripple-Effekt und rechts mit geöffneter Sprachauswahl	39
5.7	Impressum	39
5.8	Login-Dialog	40
5.9	Text und Überschrift	41
5.10	Texteingabefeld links ohne und rechts mit Eingabe	42
5.11	Eingabefeld für die Bereichsauswahl links ohne und rechts mit Eingabe .	42
5.12	Tastaturen für Text- und Zahleneingabe	43
5.13	Kalender in Portrait-Ansicht	44
5.14	Kalender in Landscape-Ansicht	45
5.15	Editor für Bereichsauswahl in Portrait-Ansicht	45
5.16	Editor für Bereichsauswahl in Landscape-Ansicht	46
5.17	Zwei- und Dreistellungsschalter links vor und rechts nach der Eingabe . .	47
5.18	Mehrstellungsschalter oben mit und unten ohne Eingabe	48
5.19	Screenshot mit aktivem Floating Action Button	48

Name: Florian Hofherr

Matrikelnummer: 790560

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Florian Hofherr